



FACULTAD DE INFORMATICA



UNIVERSIDAD
NACIONAL
DE LA PLATA

Enseñanza de Métodos ágiles de Desarrollo de Software en Argentina. Estado del Arte

Alumno: Nicolás Martín Paez

Trabajo Final de Especialización de la Maestría en Tecnología Informática
Aplicada en Educación

Abril 2020

La Plata, Argentina

Director: Lic. Alejandro Oliveros

Co-Directora: Dra. Alejandra Zangara

Para Emma

Agradecimientos

A todos los docentes que colaboraron con el relevamiento realizado en este trabajo, ya sea completando la encuesta o facilitando contactos, su colaboración fue central para la realización de esta investigación.

A Alejandro Oliveros, quien me insistió para completar mi carrera de especialización y que adicionalmente me acompañó en el proceso como director.

A Alejandra Zangara, la docente que más me movilizó cuando cursé las materias de la especialización y que luego me acompañó como co-directora en este último tramo.

A mi amigo y colega Diego Fontdevila quien estuvo acompañándome desde un primer momento cuando decidí iniciar este trabajo.

A Lucia, por su paciencia, acompañamiento y visión alternativa del mundo.

A mamá, Silvia, que siempre está.

Resumen

Los métodos ágiles, gestados a finales de los años 90', son en la actualidad parte del *mainstream* en la industria del software. Su popularidad ha alcanzado niveles muy altos llegando incluso a los ambientes académicos. A la extensa oferta de libros existentes sobre métodos ágiles se suman artículos científicos y escritos informales. Existen también muchos artículos reportando experiencias de enseñanza de métodos ágiles en distintas partes del mundo. Incluso, algunos de esos artículos relatan experiencias realizadas en Argentina a pesar de que los lineamientos vigentes del Ministerio de Educación, para las carreras del área de Informática, no hacen ninguna mención explícita a la enseñanza de métodos ágiles.

El presente trabajo presenta un estado del arte de la enseñanza de métodos ágiles en las carreras de grado de informática en las universidades argentinas. Dicho estado del arte se estableció a partir de una revisión formal de literatura y de un estudio de campo basado en una encuesta a docentes de ingeniería de software.

Los resultados obtenidos en esta investigación indican que los métodos ágiles son enseñados en el 97,7% de las carreras de grado relevadas. El 83,8% de los encuestados dicta sus materias de forma completamente presencial y una porción de los mismos implementa algún grado de hibridación en sus propuestas dado por el uso de campus virtuales. El porcentaje de uso de tecnologías de la información no resulta sorprendente considerando que son carreras tecnológicas y que sus alumnos son nativos digitales.

Índice

| | |
|---|-----------|
| Resumen | II |
| 1. Introducción | 1 |
| 1.1. Objetivos | 2 |
| 1.2. Motivación | 2 |
| 1.3. Preguntas de Investigación | 3 |
| 1.4. Metodología de investigación | 3 |
| 1.5. Terminología | 4 |
| 1.6. Organización del trabajo | 5 |
| 2. Trabajos relacionados | 6 |
| 2.1. Fuentes y estrategia de búsqueda | 8 |
| 2.2. Proceso de revisión | 11 |
| 2.3. Resultados | 12 |
| 3. Sobre el Movimiento Ágil | 16 |
| 3.1. El manifiesto ágil | 17 |
| 3.2. Más allá del manifiesto | 19 |
| 3.3. Los métodos ágiles | 19 |
| 3.4. Las prácticas ágiles | 22 |
| 4. Sobre el Proceso de Enseñanza | 28 |
| 4.1. Aprender y Enseñar | 29 |
| 4.2. Tecnologías de la Información y de la Comunicación | 32 |
| 4.3. Entornos Virtuales | 34 |
| 4.4. Modalidades de enseñanza | 35 |
| 5. Diseño del relevamiento | 38 |
| 5.1. Audiencia | 39 |
| 5.2. Implementación del cuestionario | 40 |
| 5.3. Estructura del cuestionario | 40 |
| 5.4. Evaluación preliminar del cuestionario | 41 |
| 5.5. Ejecución de la encuesta | 41 |
| 6. Resultados | 42 |
| 6.1. Presencia de los métodos ágiles en las carreras | 43 |
| 6.2. Enseñanza de métodos ágiles | 44 |
| 6.3. Estrategia de enseñanza | 46 |
| 6.4. Didáctica de la enseñanza | 48 |
| 6.5. Uso de tecnologías | 49 |
| 6.6. Análisis cruzado de las respuestas | 50 |
| 7. Discusión | 51 |
| 7.1. Sobre la representatividad de la muestra | 52 |
| 7.2. Sobre el impacto de COVID-19 | 52 |

| | |
|---|-----------|
| 8. Conclusión | 53 |
| 8.1. Sobre las preguntas de investigación | 54 |
| 8.2. Reflexiones generales | 55 |
| 8.3. Trabajos futuros | 55 |
| Referencias | 57 |
| Apéndice | 61 |
| A1. Apéndice 1 - Cuestionario | 61 |
| A2. Apéndice 2 - Participantes del relevamiento | 68 |

Índice de tablas

| | |
|---|----|
| 2.1. Buscadores de fuentes. | 10 |
| 2.2. Instituciones y publicaciones. | 12 |
| 2.3. Artículos seleccionados | 13 |
| 2.4. Publicaciones por año. | 14 |
| 2.5. Prácticas mencionadas explícitamente | 14 |
| 6.1. Estrategias didácticas dentro y fuera del aula (valores expresados en %) . | 49 |

Índice de figuras

| | |
|---|----|
| 2.1. Resultado de la etapa 1 del proceso de revisión | 11 |
| 2.2. Etapas del proceso de revisión | 12 |
| 4.1. Esquema del Modelo de Enseñanza y Aprendizaje de Pedro Hernández . | 30 |
| 4.2. Triángulo pedagógico de Houssaye | 31 |
| 4.3. Distancia Transaccional de Moore | 36 |
| 6.1. Enseñanza de métodos ágiles en las carreras | 44 |
| 6.2. Métodos ágiles enseñados | 45 |
| 6.3. Prácticas ágiles enseñadas | 45 |
| 6.4. Enseñanza teórica y práctica | 46 |
| 6.5. Modalidad de Enseñanza | 47 |
| 6.6. Modalidad de evaluación | 48 |
| 6.7. Tecnologías utilizadas en la enseñanza | 49 |

1. Introducción

Todos somos muy ignorantes. Lo que ocurre es que no todos ignoramos las mismas cosas.

Albert Einstein

1.1. Objetivos

El objetivo de este trabajo es establecer el estado del arte de la enseñanza de métodos ágiles de desarrollo de software en Argentina. En la industria del software es muy común el uso de métodos ágiles y la academia ha tomado nota de ello. Las universidades están incorporando la enseñanza de métodos ágiles a su cursos mientras que los investigadores han iniciado líneas de investigación sobre esta temática. Este trabajo se propone establecer en qué medida y de qué manera las universidades han incorporado dicha temática en Argentina. Más precisamente se pretende identificar las metodologías de enseñanza utilizadas y las tecnologías de mediación y apoyo a la enseñanza que se incorporan.

1.2. Motivación

Los métodos ágiles surgieron a finales de la década del 1990 aunque su nacimiento formal suele citarse en 2001 con la publicación del Manifiesto Ágil (Beck et al., 2001). Desde entonces la popularidad de los métodos ágiles ha ido en constante ascenso. Ya en 2010 un estudio de Forrester (West and Grant, 2010) indicaba que los métodos ágiles habían alcanzado el *mainstream*. Esta gran popularidad de los métodos ágiles surgidos en el ámbito industrial fue gradualmente permeando en la academia. En la actualidad es muy común encontrar trabajos sobre métodos ágiles en las conferencias de Ingeniería de Software y hasta hay conferencias industriales y académicas dedicadas por completo a esta temática. Los libros clásicos de Ingeniería de Software como los de Pressman (Pressman, 2014) y Sommerville (Sommerville, 2015) han incluido métodos ágiles hace ya varias ediciones. Diversos trabajos se han presentado a nivel global reportando experiencias en la enseñanza de métodos ágiles. Algunos de los mismos sugieren cierta superficialidad en el abordaje de esta temática lo cual dificulta el desempeño de los estudiantes al momento de su inserción profesional. Por otro lado, hay algunas experiencias más generales sobre la enseñanza de la Ingeniería de Software que describen el uso de estrategias no tradicionales como aula invertida y aprendizaje centrado en el alumno. Sin embargo, no hay estudios sobre el estado del arte de la enseñanza de métodos ágiles que aborden específicamente la situación en Argentina ni tampoco que analicen la forma en que estos métodos son enseñados en términos de aspectos pedagógicos/didácticos y dinámica de mediación. Por otro lado,

si bien las recomendaciones curriculares de IEEE y ACM¹ (ACM) para las carreras de ingeniería de software y tecnología de información mencionan explícitamente los métodos ágiles, las regulaciones del Ministerio de Educación de Argentina (Resolución 786/2009, 2009) sobre las carreras de informática no hacen mención alguna a esta temática. De todo esto es que surge la motivación para realizar este trabajo.

1.3. Preguntas de Investigación

Las preguntas de investigación (PI) a las que se pretende dar respuesta con esta investigación son:

- PI1: ¿Se enseñan métodos ágiles de desarrollo de software en las carreras universitarias de grado de informática en Argentina?
- PI2: ¿Cuál es el foco de estudio de los métodos ágiles (teórico, teórico/práctico, otro)?
- PI3: ¿En qué medida se utilizan tecnologías de la información en la enseñanza de métodos ágiles?
- PI4: ¿Se utiliza alguna modalidad híbrida de enseñanza como aula invertida y/o extendida?
- PI5: ¿Qué metodologías didácticas se utilizan en la enseñanza de métodos ágiles?

Al dar respuesta a estas preguntas podremos entender la situación actual en Argentina y, a partir de ello, se podrán identificar líneas de acción para futuras propuestas curriculares y pedagógicas/didácticas.

1.4. Metodología de investigación

En términos de metodología de investigación el trabajo principal consistió en un relevamiento de campo vía encuestas a docentes de ingeniería de software de diversas

¹ACM es la sigla de Association for Computing Machinery, una asociación internacional que reúne desde 1947 a científicos y educadores del área de computación. Esta asociación genera recomendaciones y lineamientos para las carreras de computación e informática.

universidades argentinas. Previo a este relevamiento de campo se realizó una revisión sistemática de literatura. Los resultados de dicha revisión sistemática fueron insuficientes para dar respuesta a todas las preguntas de investigación planteadas. Sin embargo, esta revisión fue publicada como resultado parcial del presente trabajo en el “Workshop de Innovación en Educación en Informática de la Conferencia Argentina de Ciencia de la Computación, CACIC 2019”. Asimismo, un resumen de dicha revisión es presentado en el capítulo 2 del presente trabajo.

1.5. Terminología

De acuerdo a las regulaciones del Ministerio de Educación (Resolución 786/2009, 2009) existen en Argentina cinco títulos de carreras de grado en el área de Informática / Sistemas / Computación:

- Ingeniero en Sistemas de Información / Informática
- Ingeniero en Computación
- Licenciado en Ciencias de la Computación
- Licenciado en Sistemas / Sistemas de Información
- Licenciado en Informática

En el contexto de este trabajo se utiliza el término genérico “Carreras de Informática” para referirse a las carreras que otorgan estos títulos.

Resulta muy común en la industria utilizar el término “Desarrollo Ágil” para hacer referencia al uso de métodos ágiles de desarrollo de software. En el contexto de este trabajo se utilizan ambos términos en forma indistinta.

Mucha de la terminología de la ingeniería de software se encuentra en inglés y, a pesar de que distintos autores han hecho esfuerzos de traducción, en el caso del desarrollo ágil no ha habido un consenso absoluto sobre las traducciones al castellano de ciertos términos. Es por ello que hay libros de desarrollo ágil escritos en castellano pero que igualmente utilizan la terminología original en inglés. A partir de esto, es que en el contexto de presente trabajo se optó por utilizar la terminología original en inglés.

1.6. Organización del trabajo

El resto de este trabajo está organizado de la siguiente forma: el capítulo dos presenta un resumen de trabajos relacionados que fue elaborado a partir de una revisión formal de literatura.

Los capítulos tres y cuatro abordan el marco teórico de los métodos ágiles y los procesos de enseñanza respectivamente.

Los capítulos cinco y seis explican la metodología de investigación utilizada y los resultados obtenidos a partir de la aplicación de la misma.

En el capítulo siete se discuten algunas cuestiones sobre los resultados.

Finalmente en el capítulo ocho se presentan las conclusiones y trabajos futuros.

2. Trabajos relacionados

Si he visto más lejos es porque estoy
sentado sobre los hombros de gigantes.

Isaac Newton

Al buscar trabajos sobre educación ágil se pueden encontrar resultados pertenecientes a dos categorías distintas que a simple vista pueden parecer lo mismo pero que no lo son. Una de las categorías se refiere la enseñanza de métodos ágiles de desarrollo de software. La otra aborda el uso de técnicas ágiles para enseñar alguna temática que puede incluso exceder el desarrollo de software. Este trabajo está enfocado en la primera categoría: la enseñanza de métodos ágiles de desarrollo de software.

A nivel mundial existen varios estudios sobre la enseñanza de Desarrollo Ágil. En Finlandia, Paasivaara et al. (Paasivaara et al., 2014) reportaron un experiencia utilizando bloques Lego para enseñar *Scrum*. En Brasil, Von Wangenheim et al. (Gresse von Wangenheim et al., 2013) desarrollaron un juego educativo llamado Scrumia para enseñar *Scrum* en los cursos de computación. Persson et al. (Persson et al., 2012) propusieron una adaptación de *Scrum* para ajustarlo a las necesidades particulares de los contextos de la enseñanza universitaria en Suecia. Kropp y Meier (Kropp and Meier, 2013) estudiaron la enseñanza de *Agile* en Suiza y entre sus conclusiones resaltan la importancia de integrar el desarrollo ágil no solo conceptualmente sino también llevándolo a la práctica y en base a esto hacen una propuesta de enseñanza. Hay también varios trabajos reportando el uso de métodos ágiles en el contexto de trabajos finales de carrera (Rico and Sayani, 2009; Mahnic, 2012; Lu and Declue, 2011).

Por otro lado, existen algunos estudios sobre la evolución y el estado del arte de la enseñanza de métodos ágiles en regiones particulares, lo cual está en sintonía con el objetivo de este trabajo. Chookittikul et al (Chookittikul et al., 2011) llevaron a cabo un estudio en Tailandia en el cual realizaron entrevistas en las universidades líderes del país para entender como los métodos ágiles fueron incorporados en las carreras de informática. Asimismo, un amplio estudio sobre la evolución de *Agile* fue realizado en Brasil por Melo (Melo et al., 2013), el mismo cubrió los campos de industria, educación e investigación.

Respecto de la situación en Argentina, hay varios trabajos que relatan experiencias y/o propuestas de enseñanza de métodos ágiles en las carreras de informática. Sin embargo, al momento de comenzar con esta investigación no había publicado ningún trabajo que describiera de manera integral el estado de la enseñanza de métodos ágiles en universidades argentinas. Esta situación es la que motivó la realización de este trabajo.

Con esta motivación se planteó inicialmente realizar un estudio de tipo mapeo sistemático

el cual implica un proceso de revisión formal de publicaciones. Dicha revisión resultó insuficiente para dar respuesta a las preguntas de investigación planteadas inicialmente en este trabajo. Es por eso que esa revisión se convirtió en un hito intermedio de esta investigación publicándose en el “Workshop de Innovación en Educación en Informática en el Congreso Argentino de Ciencias de la Computación, CACIC 2019” (Paez et al., 2019). En las siguientes secciones se presentan de forma resumida los puntos destacados de dicho trabajo de revisión.

El trabajo de revisión fue realizado siguiendo los lineamientos de Kitchenham (Kitchenham and Charters, 2007) y Kuhrmann (Kuhrmann et al., 2017b) para la elaboración de mapeos sistemáticos.

2.1. Fuentes y estrategia de búsqueda

En Argentina existen siete conferencias académicas nacionales en el área de informática.

- CACIC: Congreso Argentino de Ciencias de la Computación.
- JAIIO: Jornadas Argentinas de Informática e Investigación Operativa.
- TEyET: Congreso Nacional de Tecnología en Educación y Educación en Tecnología.
- WICC: Workshop de Investigadores en Ciencias de la Computación.
- CONAIISI: Congreso Nacional de Ingeniería Informática - Sistemas de Información.
- CACIDI: Congreso Argentino de Ciencias de la Informática y Desarrollos de Investigación.
- ARGENCON: Congreso Bienal de IEEE Argentina.

Adicionalmente existe un conjunto de revistas relevantes en este área:

- JCS&T: Journal of Computer Science & Technology
- Revista TEYET: Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología
- EJS: Electronic Journal of SADIO

Para buscar en todas estas fuentes se utilizaron distintos motores de búsqueda porque no existía uno que cubriera todas estas fuentes.

La cadena de búsqueda estuvo compuesta por dos partes conectadas por un operador 'AND':

1. agile OR ágil OR Scrum
2. educación OR education OR enseñanza OR teaching

En la primera parte de la cadena de búsqueda se decidió explícitamente incluir el término *Scrum* porque se consideró que su gran difusión en la industria ha generado que en ocasiones los practicantes se refieran a *Scrum* como sinónimo de *Agile* (Tranter, 2018) incluso cuando esto no es correcto ya que Scrum no es más que un método ágil dentro del ecosistema *Agile*.

La cadena de búsqueda resultante quedó conformada como:

(1) AND (2)

Al intentar esta búsqueda en los distintos buscadores se encontraron varias restricciones y diferencias entre ellos. Por ello es que si bien conceptualmente en todos los casos se utilizó la cadena de búsqueda previamente mencionada, en términos de implementación dicha cadena tuvo que ser ajustada de acuerdo a las limitaciones de cada buscador lo cual en algunos casos implicó la ejecución de varias búsquedas parciales.

Para buscar en las memorias de CACIC, JAIIO, TEYET, JJCS&T y WIIC, se utilizó SEDICI ², el repositorio institucional de la Universidad Nacional de La Plata, que más allá de las publicaciones de sus miembros también almacena artículos de otras instituciones como Red UNCI (la institución que organiza CACIC, TEYET y WICC) y SADIO (la institución que organiza JAIIO). Cabe destacar que SEDICI tiene artículos publicados en JAIIO desde 2013, es por ello que las publicaciones de JAIIO previas a 2013 no fueron cubiertas en este trabajo. SEDICI no soporta varios conectores en la cadena de búsqueda pero ofrece la posibilidad de filtrar los resultados de la búsqueda por palabra clave y temática. Es por ello que al buscar en SEDICI se debieron realizar varias búsquedas parciales y luego filtrar utilizando las palabras clave y la temática.

²<http://sedici.unlp.edu.ar/>

Para buscar en los artículos de CACIDI y ARGENCON se utilizó IEEE Xplore ³ debido a que estas dos conferencias pertenecen a IEEE. En estos casos la cadena de búsqueda fue extendida para agregar el nombre de la conferencia (CACIDI o ARGENCON en cada caso).

Para buscar en publicaciones internacionales se utilizó Scopus ⁴ que cubre las conferencias y revistas más relevantes del área de educación e ingeniería de software como son: *International Conference in Software Engineering*, *Computer and Education Journal* y *Transactions on Education Journal*. Este motor de búsqueda permite agregar restricciones a la cadena de búsqueda para limitar los resultados por el país de filiación de los autores, de esa forma se restringió la búsqueda a Argentina.

La búsqueda en las memorias de CONAIISI se realizó de forma manual ya que no son cubiertas por ningún buscador. Al mismo tiempo las distintas ediciones de CONAIISI han tomado distintas estrategias respecto de la publicación de las memorias. Las ediciones 2013, 2016 y 2018 publicaron sus memorias en sus correspondientes sitios web con un link individual a cada uno de los artículos. Por otro lado las ediciones 2014 y 2017 ofrecen un link a un único archivo que contiene todos los artículos. Finalmente de la edición 2015 no hay memorias publicadas disponibles y por ello esa edición no ha sido cubierta en este estudio.

Finalmente se realizó una búsqueda manual en las memorias del Electronic Journal de SADIO (EJS) cuya primera edición fue publicada en 1998.

La tabla 2.1 resume la forma de búsqueda en cada fuente.

Tabla 2.1: Buscadores de fuentes.

| Buscador | Fuente |
|-----------------|--|
| SEDICI | CACIC, WICC, JAIHO, JJCS&T, TEYET (congreso y revista) |
| IEEE Explore | CACIDI, ARGENCON |
| Scopus | Publicaciones internacionales |
| Búsqueda manual | CONAIISI, EJS |

Los estudios incluidos en la revisión fueron aquellos que describían o mencionaban la enseñanza o uso de técnicas de desarrollo ágil de software en carreras de grado del área de

³<https://ieeexplore.ieee.org/>

⁴<https://www.scopus.com/>

informática en universidades argentinas. Fueron excluidos aquellos estudios que trataban del uso de técnicas ágiles para la enseñanza (sea esta de desarrollo de software o no). Otros estudios excluidos fueron aquellos que proponían la enseñanza de métodos ágiles pero que no tenían una experiencia asociada que los validara. Finalmente se descartaron también los trabajos que mencionaban el uso de *agile* para desarrollar software educativo.

2.2. Proceso de revisión

El proceso de revisión constó de tres etapas. En la primera etapa, se recolectaron los artículos a partir de ejecutar las correspondientes búsquedas en las diferentes fuentes. El resultado obtenido de cada fuente fue registrado en un hoja cálculo distinta porque cada motor de búsqueda ofrecía distintas capacidades de exportación. La figura 2.1 resume la cantidad de publicaciones encontradas en cada fuente.

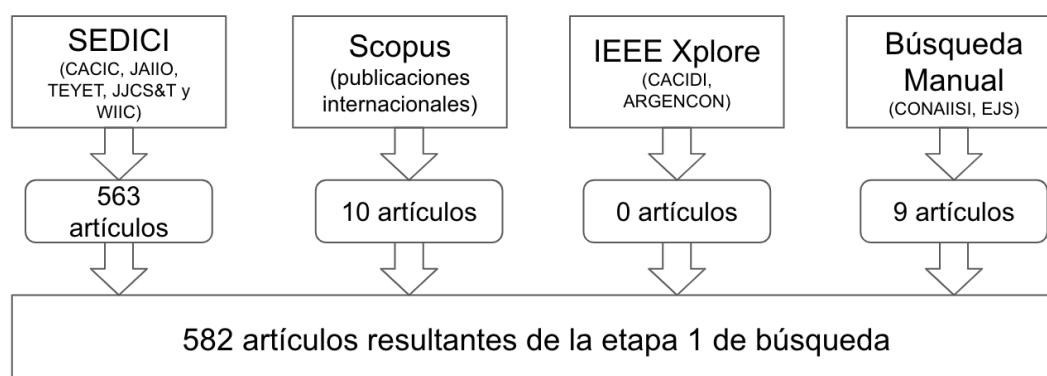


Figura 2.1: Resultado de la etapa 1 del proceso de revisión

En la segunda etapa se filtraron los artículos en base a aplicar los criterios de inclusión/exclusión sobre título y resumen de cada uno. Al final de esta segunda etapa quedaron 40 artículos todos cargados en una misma planilla facilitando la eliminación de duplicados en el paso posterior.

En la tercera etapa, se revisaron los artículos en detalle leyendo cada artículo en forma completa y extrayendo datos para el posterior análisis. En esta etapa se eliminaron los artículos duplicados y se detectó un tipo particular de publicación en las memorias de JAIIO: las comunicaciones orales. Estas publicaciones son resúmenes de artículos que ya habían sido publicados en alguna conferencia previa. A los fines de este trabajo, las

comunicaciones orales fueron consideradas como duplicados.

La figura 2.2 resume el proceso completo de revisión.

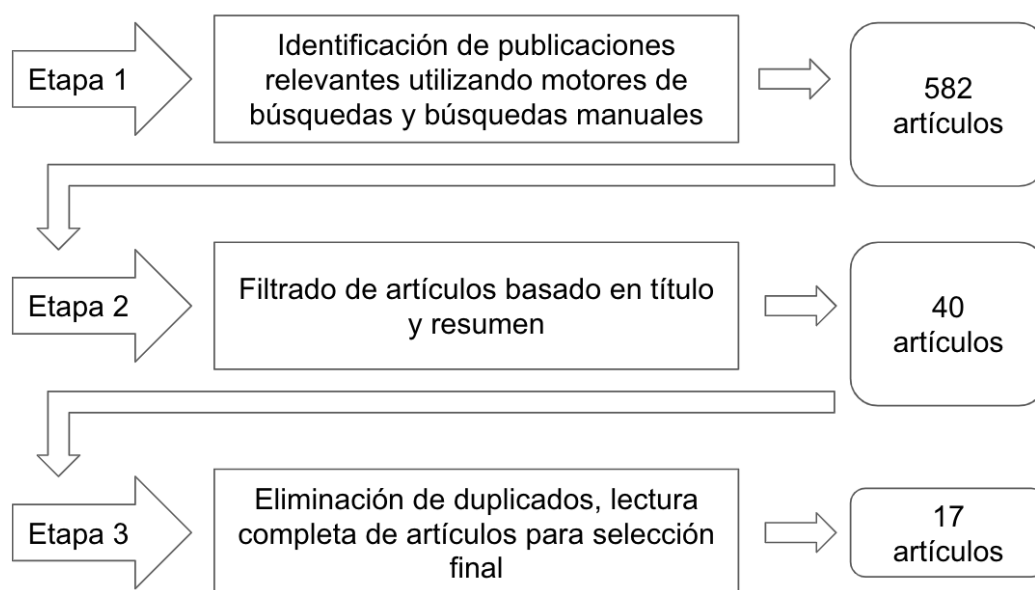


Figura 2.2: Etapas del proceso de revisión

2.3. Resultados

Las búsquedas se realizaron en junio de 2019 y se identificaron un total de 17 artículos que cumplieron con los criterios previamente descriptos respecto de la enseñanza de desarrollo ágil. Estos artículos que se encuentran listados en la tabla 2.3 pertenecen a nueve instituciones distintas, ocho de las cuales son universidades públicas (tabla 2.2).

Tabla 2.2: Instituciones y publicaciones.

| Institución | Cantidad de publicaciones |
|---------------------------|---------------------------|
| UNICEN | 8 |
| UNLAM | 2 |
| UTN Cordoba | 1 |
| U. de la Cuenca del Plata | 1 |
| UN de Salta | 1 |
| UN del Nordeste | 1 |
| UN de La Plata | 1 |
| UN de Rio Cuarto | 1 |
| UN de Tres de Febrero | 1 |

Tabla 2.3: Artículos seleccionados

| Id | Título |
|----|---|
| 1 | La Evolución de las Cátedras para Influir en los Knowmads y Formar al Trabajador de la Industria 4.0 (Burghardt et al., 2017) |
| 2 | Experiencia sobre desarrollo con Genexus siguiendo prácticas ágiles (Gimson Saravia and Gil, 2016) |
| 3 | Mejora continua aplicada en la enseñanza de la Ingeniería del Software (Ferraro et al., 2015) |
| 4 | Implementación practica del agilismo en proyecto de Ingeniería de Software (Levy et al., 2016) |
| 5 | Experiences in software engineering education: Using scrum, agile coaching, and virtual reality (Scott et al., 2017) |
| 6 | Propuesta para documentar trabajos finales utilizando metodologías ágiles (Uva et al., 2014) |
| 7 | Are learning styles useful indicators to discover how students use Scrum for the first time? (Scott et al., 2014) |
| 8 | El rol del estilo de aprendizaje en la enseñanza de prácticas de Scrum: un enfoque estadístico (Scott et al., 2013) |
| 9 | Measuring the Impact of Agile Coaching on Students' Performance (Rodríguez et al., 2016) |
| 10 | Supporting assessment of practices in software engineering courses (Rodríguez et al., 2015a) |
| 11 | Teaching Scrum to software engineering students with virtual reality support (Rodríguez et al., 2012) |
| 12 | Towards better Scrum learning using learning styles (Scott et al., 2016) |
| 13 | Virtual Scrum: A teaching aid to introduce undergraduate software engineering students to Scrum (Rodríguez et al., 2015b) |
| 14 | Mejoras en el proceso de enseñanza / aprendizaje de la programación utilizando metodologías de la industria del software (Aubin et al., 2013) |
| 15 | Metodologías activas y corrección por rúbricas en el proceso de enseñanza / aprendizaje de programación (Aubin et al., 2014) |
| 16 | A flipped classroom experience teaching software engineering (Paez, 2017) |
| 17 | ¿Como Pueden Scrum y las TICs Mejorar el Aprendizaje Colaborativo en el ámbito Universitario? (Lanza Casteli, 2017) |

Adicionalmente a los artículos seleccionados, se encontró un conjunto de artículos que, si bien no cumplieron con el criterio de inclusión, resulta relevante mencionarlos: a) un

artículo proponiendo un plan de estudios para una carrera de pre-grado que incluía explícitamente métodos ágiles, b) dos artículos describiendo cursos extra-curriculares sobre Desarrollo ágil de Software.

La tabla 2.4 muestra la cantidad de publicaciones por año, como puede observarse la primera publicación encontrada data del año 2012. Cabe destacar que si bien el nacimiento formal de los métodos ágiles se sitúa en 2001, no se utilizó este hecho como criterio explícito de selección.

Tabla 2.4: Publicaciones por año.

| Año | Cantidad de publicaciones |
|------|---------------------------|
| 2012 | 1 |
| 2013 | 1 |
| 2014 | 5 |
| 2015 | 3 |
| 2016 | 5 |
| 2017 | 4 |
| 2018 | 1 |

El método ágil más mencionado en los artículos fue *Scrum* (mencionado en 11 artículos) seguido por *Extreme Programming* (mencionado por 3 artículos). Esta popularidad de *Scrum* está en línea con lo que ocurre en la industria (Version One, 2019). Un artículo hace mención explícita a *Scrum* y *Extreme Programming* simultáneamente. Dos artículos no mencionan ningún método ágil en particular. La mayoría de los artículos no hacen mención a prácticas ágiles más allá de las propuestas por *Scrum*. La tabla 2.5 detalla las menciones de prácticas por artículo.

Tabla 2.5: Prácticas mencionadas explícitamente

| Prácticas | Artículos |
|-------------------------|-----------|
| Self-Organization | 2, 9 |
| Definition of Done | 2 |
| Automated testing | 9 |
| Continuous Integration | 9 |
| Release Planning | 12 |
| Planning Poker | 12 |
| Pair-Programming | 14, 15 |
| Test-Driven Development | 16 |
| Story Mapping | 16 |
| Retrospectives | 16 |

Respecto de la forma de enseñanza, un tercio de los artículos no hace ninguna mención. Entre los artículos que sí mencionan algo sobre la estrategia de enseñanza tenemos:

- Los artículos 1 y 15 mencionan un enfoque de aprendizaje basado en problemas.
- Los artículos 5, 11 y 13 mencionan el uso de realidad virtual (nótese que los tres artículos pertenecen al mismo grupo de autores).
- Los artículos 7, 8 y 12 mencionan los estilos de aprendizaje de Felder-Silverman.
- El artículo 16 menciona el enfoque de aula invertida.
- El artículo 17 menciona un enfoque de aprendizaje colaborativo.

Cabe destacar que en muchos casos se les pide a los estudiantes el desarrollo de un proyecto en el contexto de un curso de Ingeniería de Software lo cual puede interpretarse como una estrategia del tipo “aprender haciendo”.

La evidencia recolectada en el trabajo de revisión aquí descripto muestra que los métodos ágiles están presentes en las carreras de informática de nueve universidades argentinas. Al mismo tiempo *Scrum* se muestra como el método ágil más difundido al ser mencionado en 12 de las 17 publicaciones encontradas. La información encontrada respecto del enfoque de enseñanza es poca y varios de los artículos no hacen ninguna mención al respecto. Los artículos que si mencionan la cuestión no tienen un enfoque unificado pero la estrategia de “aprender haciendo” es mencionada en más de un artículo.

3. Sobre el Movimiento Ágil

Un gran poder conlleva una gran
responsabilidad.

Peter Parker

3.1. El manifiesto ágil

El movimiento ágil surgió a finales de la década del 1990 aunque su nacimiento formal suele citarse en 2001 cuando un conjunto de profesionales reconocidos del desarrollo de software se reunieron en la localidad de Snowbird, Estados Unidos. Los participantes de esa reunión ya venían trabajando de formas alternativas a lo que marcaba el *mainstream* por aquella época. Algunas personalidades destacadas de esa reunión fueron Kent Beck, organizador de la reunión y creador del enfoque ágil *Extreme Programming*; Alistair Cockburn, creador de la familia de métodos ágiles *Crystal* y Ken Schwaber y Jeff Sutherland, autores del marco de trabajo *Scrum*, el enfoque ágil más popular en la actualidad. Como resultado de esa reunión se publicó el Manifiesto por el Desarrollo Ágil de Software (Beck et al., 2001). Este manifiesto ágil es concretamente una declaración de 4 valores y 12 principios que si bien surgieron en el contexto del desarrollo de software expresan una filosofía de trabajo que excede dicho contexto y que en forma más general se la suele denominar como *agile*. El recuadro 3.1 contiene los valores del Manifiesto ágil mientras que el recuadro 3.2 contienen sus principios.

Recuadro 3.1 - Valores del Manifiesto Ágil

Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

- Individuos e interacciones sobre procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual
- Respuesta ante el cambio sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

Recuadro 3.2 - Principios del Manifiesto Ágil

1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
7. El software funcionando es la medida principal de progreso.
8. Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Desde la publicación del manifiesto la popularidad de *agile* ha ido en constante ascenso. Un artículo de Forrester (West and Grant, 2010) del año 2010 ya considera *agile* como *mainstream*. Esta popularidad de *agile* ha llevado a que, en los últimos años, haya habido una expansión de su uso a diversos contextos como ser recursos humanos (*Agile HR*) y educación (*Agile Education*).

3.2. Más allá del manifiesto

Al hablar de métodos ágiles nos estamos refiriendo obviamente al conjunto de valores y principios propuesto por el manifiesto pero también a todo un conjunto de métodos y prácticas de desarrollo.

Un método define: un proceso, un conjunto de roles, prácticas y artefactos, para llevar adelante un proyecto con el objetivo de generar un producto o brindar un servicio.

Al mismo tiempo, hay que destacar que, si bien cada método define un conjunto de prácticas, es posible utilizar una o varias prácticas específicas sin usar un método completo. Más aún, en los últimos años ha cobrado relevancia la idea de “métodos híbridos” (Kuhrmann et al., 2017a) en referencia a una situación muy común en la industria en la actualidad que es el hecho de no utilizar un método siguiendo estrictamente sus definiciones sino de mezclar propuestas de más de un método.

No existe un manual que nos diga si un método de trabajo es ágil o no, entonces para saber si un método es ágil debemos cotejarlo con los valores y principios del manifiesto. Obviamente esta heurística puede resultar subjetiva, pero independientemente de eso hay un conjunto de métodos que todo el mundo reconoce como métodos ágiles.

De los valores y principios del manifiesto se desprenden un conjunto de corolarios como ser:

- Equipos pequeños y auto-organizados
- Trabajo en forma iterativa e incremental
- Entrega constante de valor a un ritmo sostenible
- Mejora continua

3.3. Los métodos ágiles

Dentro de los métodos ágiles encontramos un continuo de métodos que oscilan entre dos extremos. En un extremo están los métodos enfocados exclusivamente en la gestión, en el otro, están los métodos que más allá de la gestión ponen un foco importante en

las cuestiones de ingeniería/programación. A continuación se ofrece una breve reseña de algunos de los métodos ágiles más conocidos.

3.3.1. Scrum

Los orígenes de *Scrum* se sitúan de mediados de los 80' en el campo de la manufactura pero fue recién a mediados de los noventa cuando Jeff Sutherland y Ken Schwaber desarrollaron formalmente *Scrum* en desarrollo de software (Schwaber, 1997). *Scrum* tal como fue definido es un marco de trabajo ya que hace una propuesta metodológica intencionalmente incompleta que debe ser completada por quien pretenda trabajar con *Scrum*. A modo de ejemplo: *Scrum* indica que el trabajo será llevado adelante por un equipo multidisciplinario pero no entra en detalle sobre las habilidades específicas de los miembros de ese equipo. Dentro del continuo de los métodos ágiles *Scrum* está claramente del lado de los métodos de gestión. Esto es así a punto tal que en los últimos años es común ver equipos trabajando con *Scrum* fuera del área del desarrollo de software. Actualmente, y desde hace ya varios años, *Scrum* es el método ágil más popular y más utilizado. Esta popularidad ha llevado a que se desarrollen variantes de *Scrum* para entornos educativos como EduScrum (Wijnands and Stolze, 2019).

3.3.2. Extreme Programming

Extreme Programming (XP) es una metodología de desarrollo de software creada por Kent Beck a fines de los años 90' y documentada inicialmente en su libro *Extreme Programming Explained* (Beck, 1999). En el continuo de los métodos ágiles Extreme Programming está en el extremo de los métodos enfocados en la ingeniería.

A diferencia de *Scrum*, XP es explícito en toda su propuesta, no sólo indica qué cuestiones debe realizar el equipo sino que también indica cómo deben ser realizadas. A modo de ejemplo: XP indica concretamente los distintos roles y habilidades que conforman el equipo de desarrollo que llevará adelante la construcción.

Varias de las prácticas desarrollo del mundo *agile* tuvieron su origen en *Extreme Programming*. Prácticas como *Pair-Programming*, *Refactoring*, *Continuous Integration* y

Test-Driven Development forman parte de la definición inicial de *Extreme Programming*, pero han logrado una transcendencia tal que es común el uso de estas prácticas independientemente del uso de XP.

3.3.3. Agile Unified Process

El Proceso Unificado (UP por sus siglas en inglés) es una metodología iterativa e incremental creada y popularizada durante los años 90. La versión original fue desarrollada con el apoyo de la empresa Rational y por ello es conocida como *Rational Unified Process*. Con el correr de los años fueron surgiendo variantes y adaptaciones entre las que se encuentra el *Agile Unified Process* cuyo principal impulsor es Scott Ambler. La propuesta del *Agile Unified Process* simplifica lo propuesto por Rational UP e incorpora algunas prácticas adicionales del mundo ágil como *Test-Driven Development*.

3.3.4. Kanban

Kanban (Anderson, 2013) es una forma de diseñar, administrar y mejorar sistemas de flujo de trabajo. Viene inspirado en la filosofía de producción *Lean*, originada en Toyota hace ya varias décadas. Para algunos resulta exagerado considerar a Kanban un método por el hecho de que en términos concretos solo propone un puñado de reglas para reflejar/asegurar los principios de la filosofía *Lean*. El uso de Kanban en el contexto de desarrollo de software fue impulsado por David J. Anderson alrededor de 2005. El hecho de mencionarlo como un método ágil tiene ver con que conceptualmente los principios *Lean* (y de Kanban por extensión) están muy alineados, lo que ha llevado a que Kanban sea comúnmente utilizado en contextos ágiles.

3.3.5. Crystal

Crystal es una familia de métodos desarrollada por Alistair Cockburn a mediados de los años '90 (Cockburn, 2004). Cada una de las variantes de *Crystal* se identifica con un color y está enfocada en un tamaño de equipo particular. Más allá de las particularidades de

cada variante de *Crystal*, hay un conjunto de elementos comunes característicos a todas ellas:

- Entrega frecuente
- Mejora reflexiva (*Reflective improvement*)
- Comunicación osmótica
- Seguridad personal (*Personal safety*)
- Foco
- Fácil acceso a usuarios expertos
- Ambiente técnico con pruebas automatizadas, gestión de la configuración e integración frecuente

3.3.6. Agile a Escala

Entre las características comunes que comparten los métodos ágiles está el trabajo en equipos pequeños, típicamente no más de 9 personas por equipo. Sin embargo, en ocasiones el trabajo a realizar tiene una dimensión tal que sumado a ciertas restricciones de tiempo resulta imposible para un equipo pequeño llevarlo a cabo. Es en ese contexto que en los últimos años comenzaron a popularizarse diversas propuestas de marcos de trabajo para poder llevar adelante el desarrollo de sistemas de mayor tamaño utilizando valores, principios y prácticas ágiles. Algunas de esas propuestas son *Disciplined Agile* (Ambler and Lines, 2012), SAFe (Leffingwell, 2018) y LeSS (Larman and Vodde, 2016). Si bien los enfoques de cada propuesta son distintos, todas apuntan a como organizar el trabajo de N equipos en el contexto de una organización.

3.4. Las prácticas ágiles

Como ya se mencionó, la mayoría de las prácticas tienen su origen en algún método en particular. Sin embargo, muchas prácticas pueden utilizarse en forma independiente de los métodos que las originaron (aunque tal vez con menor efectividad). Esta independencia

de las prácticas impacta positivamente tanto en el proceso de adopción de prácticas como también en el proceso de aprendizaje ya que permite un abordaje más granular.

En forma similar a lo que ocurre con los métodos, las prácticas también pueden categorizarse. Una categorización posible es: prácticas técnicas o prácticas de gestión (Paez et al., 2018). Las prácticas técnicas son aquellas que tienen que ver concretamente con la forma en que se construye software y que justamente por ello solo tienen sentido en el contexto del desarrollo software. Siguiendo, las prácticas de gestión tienen que ver con la forma en que se organiza el trabajo y no están necesariamente asociadas a la construcción de software, habilitando a que dichas prácticas pueden ser utilizadas en cualquier contexto de organización de trabajo. *Test-Driven Development* y *Daily Stand-ups* son claros ejemplos de prácticas técnicas y de gestión respectivamente.

En las siguientes secciones se ofrece una breve reseña de un conjunto de prácticas ágiles. Cabe mencionar que, si bien existen más prácticas que las que aquí mencionadas, esta selección obedece al hecho de que estas prácticas son citadas tanto en reportes industriales (Version One, 2019) como en trabajos académicos (Kuhrmann et al., 2017a).

3.4.1. Test-Driven Development (TDD)

Esta práctica es parte del núcleo de *Extreme Programming*, fue creada por el mismo Kent Beck. Es, posiblemente, una de las prácticas más famosas del mundo *agile* pero es al mismo tiempo una de las prácticas con menor uso. La relación entre fama y uso en el caso de TDD no tiene igual: no hay práctica con el nivel de fama de TDD y que al mismo tiempo tenga tan poco uso. Esto posiblemente se deba que por un lado es una práctica muy efectiva y fácil de entender pero muy difícil de aplicar.

3.4.2. Pair-Programming

Esta es otras de las prácticas del núcleo de XP. La idea central de la práctica es bastante simple: programar de a pares, básicamente dos programadores trabajando simultáneamente en una única computadora.

A pesar de su simplicidad y de sus demostrados beneficios esta práctica no es tan utilizada

como se podría esperar. Al mismo tiempo, una parte importante de quienes la utilizan lo hacen con la intención de entrenar personas a partir de poner a trabajar juntos un programador experimentado con otro sin experiencia o de menor experiencia. Si bien este es un uso válido de la práctica no es la intención central de la misma. Lo que busca la práctica es propiciar la propiedad colectiva del código y llevar al máximo los beneficios que muchos años antes ya demostraron las revisiones de código.

3.4.3. User Stories

Las *user stories* son un artefacto que se utiliza en los contextos *agile* para representar las funcionalidades de la aplicación a construir. En su forma más pura suele decirse que una *user story* no es más que un recordatorio de una conversación que debe tener el equipo de desarrollo con el usuario sobre el tema de la *user story*.

3.4.4. Planning Poker

A pesar de su nombre esta no es una práctica de planificación sino de estimación. Más precisamente es una técnica de estimación que está dentro del grupo de las denominadas técnicas basadas en expertos aunque no requiere que quienes hagan la estimación sean expertos. En términos muy generales la técnica tiene 2 premisas:

- Los estimadores son quienes realizarán el trabajo.
- Se utilizan cartas para evitar que los estimadores se influencien mutuamente al dar sus estimaciones.

La técnica *Planning Poker* es una variante de la famosa técnica *Delphi*.

3.4.5. Retrospectivas

Esta es una de las prácticas más populares y más utilizadas del mundo *agile* tal vez por ser parte de Scrum. La retrospectiva es una reunión que se realiza periódicamente, típicamente al final de cada iteración, donde el equipo reflexiona sobre su forma de trabajo de cara a identificar oportunidades de mejora. Esta práctica está íntimamente relacionada

con el principio número 12 del manifiesto ágil: “A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia”.

3.4.6. Continuous Integration

La práctica de *Continuous Integration* es una práctica de carácter técnico. Conceptualmente propone que al menos una vez al día el producto sea integrado. Entre otras cuestiones, esta práctica implica que todo el código fuente de la aplicación sea almacenado en un repositorio compartido. La práctica de *Continuous Integration* viene a mitigar un problema con nombre propio: el infierno de la integración (en inglés el *merge hell*)

3.4.7. Continuous Delivery

Esta es una práctica relativamente nueva que se popularizó a partir del año 2010 cuando fue publicado el libro *Continuous Delivery* de Humble y Farley. Conceptualmente *Continuous Delivery* es la habilidad de entregar en manos del usuario cambios de todo tipo (funcionalidades, *bug fixes*, ajustes de configuración, etc.) de forma rápida, segura y sostenible. La implementación de *continuous delivery* trae implícita la implementación de otras prácticas como ser Automatización de pruebas y Control de la configuración y *Continuous Integration*, entre otras.

3.4.8. Release Planning

La práctica de *Release Planning* es, como su nombre lo indica, una práctica de planificación, liviana y adaptativa. En un punto no es una práctica que resulte muy novedosa en la actualidad pero años atrás, al momento del surgimiento de los métodos ágiles, estaba muy de moda hacer planes muy detallados y bastante rígidos. La idea de esta práctica es hacer planificación constante, teniendo foco en el corto plazo e ir refinando los ítems futuros del plan a medida que se avanza.

3.4.9. Iteration Planning

Esta práctica surge al trabajar en forma iterativa. La idea es que al comienzo de cada iteración el equipo planifica el trabajo para la iteración que está por comenzar. Esta reunión de planificación suele tener una parte estratégica y otra táctica. En la parte estratégica el equipo en conjunto con el cliente/usuario define qué funcionalidades se desarrollarán. En la parte táctica el equipo define cuestiones del “cómo”, o sea cuestiones de implementación a nivel organización de tareas.

3.4.10. Iteration Review

Esta práctica también viene de la mano del desarrollo iterativo. La *iteration review* es una reunión que se realiza al final de cada iteración donde se presenta y revisa el trabajo realizado por el equipo.

3.4.11. Daily Standup

Esta es una prácticas más difundidas de *agile* tal vez por su simplicidad. La *Daily Standup* es una reunión que mantiene el equipo de desarrollo todos los días con el fin de sincronizar el trabajo. Cada miembro del equipo cuenta en qué estuvo trabajando desde la reunión anterior, en que planea trabajar a continuación e indica si tuvo alguna complicación/impedimento. Esta reunión es de corta duración, no mas de 15 minutos. De hecho, el nombre de la reunión refleja la intención de que la misma se haga de parado (*standup*) para así generar cierta incomodidad en los participantes y que la reunión no se estire innecesariamente.

3.4.12. Collective Ownership

Esta es otra de las prácticas del núcleo de XP. Lo que propone es que todo miembro del equipo puede contribuir en cualquier aspecto del proyecto. Nadie es dueño del código porque todos los miembros lo son. De esta forma se evita dependencia de una persona en particular que pueda convertirse en un cuello de botella.

3.4.13. Automatización de pruebas

La automatización de pruebas no es algo exclusivo de *agile*, de hecho es una práctica previa al surgimiento de los métodos ágiles. Lo que ocurre es que en un contexto ágil esta práctica toma mayor relevancia.

4. Sobre el Proceso de Enseñanza

La educación ayuda a la persona a
aprender a ser lo que es capaz de ser.

Hesíodo

4.1. Aprender y Enseñar

A diferencia de lo que ocurre en la ingeniería donde contamos con definiciones precisas y muchas veces universales, el mundo de la Educación se caracteriza por contar con múltiples y alternativas definiciones incluso para ciertas cuestiones importantes de la disciplina. Los conceptos de enseñanza y aprendizaje son protagonistas centrales de la Educación y son, al mismo tiempo, víctimas de esas múltiples definiciones. Al mismo tiempo, estos dos conceptos han ido variando a lo largo del tiempo. Claro está que ambos guardan íntima relación pero la ocurrencia de uno no implica la ocurrencia del otro: que el docente enseñe un tema no implica que el estudiante lo aprenda.

Aprender es un fenómeno interno que se produce a partir de procesos de significación y resignificación. Influyen en este proceso diversos factores como el tipo de contenido a aprender, la metodología didáctica y la modalidad de enseñanza. Al mismo tiempo, el enseñar es un proceso externo al sujeto al que se le está enseñando. Se desprende de esto que aún cuando quien enseña y quien aprende estén física y temporalmente en el mismo lugar no hay garantía de que el aprendizaje ocurra en el mismo momento que la enseñanza. Esta característica del proceso dual de aprendizaje y enseñanza habilita la existencia de distintos modelos y, más aún, se ve potenciada exponencialmente por la introducción de las tecnologías de la información.

Debemos tener presente que un modelo es una representación parcial de la realidad que supone cierto alejamiento de la misma con el fin de facilitar su entendimiento, destacando algunos aspectos de la realidad que considera más relevantes. En términos de Sacristán (Sacristán, 1983): “el modelo es un esquema mediador entre la realidad y el pensamiento”. Existen en la actualidad diversos modelos de enseñanza, cada uno ofrece una representación distinta de la realidad, haciendo que uno u otro modelo sea potencialmente más adecuado a uno u otro contexto. Al mismo tiempo, los modelos no deben verse como excluyentes sino al contrario: puede resultar beneficioso verlos como complementarios.

Según lo plantea Hernández (Hernández, 1989) enseñar es “el acto por el que un profesor (docente), muestra o suscita contenidos educativos (conocimientos, hábitos y habilidades), a un estudiante, a través de unos medios, en función de unos objetivos y dentro de un contexto”. Esto introduce un nuevo actor central en el proceso de enseñanza y aprendizaje:

los objetivos. Según el mismo autor aprender “es el acto por el que el estudiante intenta captar y elaborar los contenidos expuestos por el profesor, o por cualquier otra fuente de información”. El estudiante alcanza el aprendizaje a través de medios, en función de los objetivos de aprendizaje y dentro de un contexto. De esta forma se completa el modelo de enseñanza y aprendizaje propuesto por Hernández:

- Docente: quien pretende enseñar.
- Estudiante: quien pretende aprender.
- Objetivos: la razón por la cual se lleva a cabo el proceso de enseñanza.
- Contenidos: la información que se pretenden transmitir o desarrollar en el estudiante.
- Medios y estrategias: técnicas utilizadas en el proceso de enseñanza y aprendizaje.
- Contexto: en el cual ocurre el proceso de enseñanza y aprendizaje.

Hernández esquematiza su propuesta de modelo con la figura 4.1.

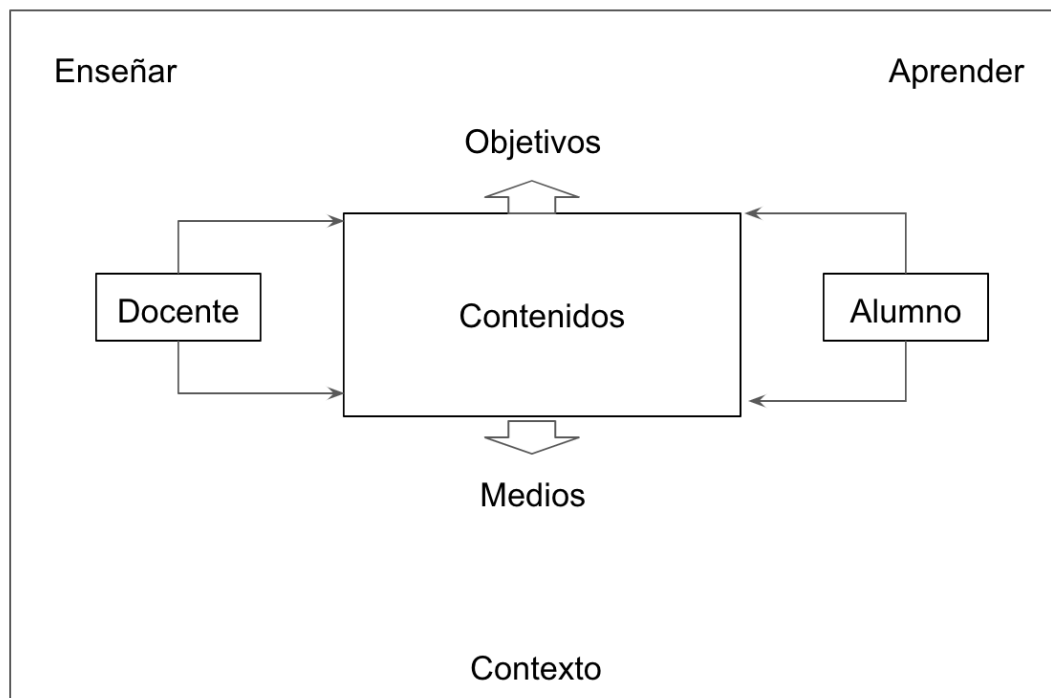


Figura 4.1: Esquema del Modelo de Enseñanza y Aprendizaje de Pedro Hernández

Un modelo de enseñanza y aprendizaje muy extendido es el denominado “modelo triangular” o “triángulo didáctico” formulado inicialmente por Jean Houssaye (Houssaye, 1984) y

reformulado posteriormente por otros autores. La figura 4.2 esquematiza los componentes de este modelo.

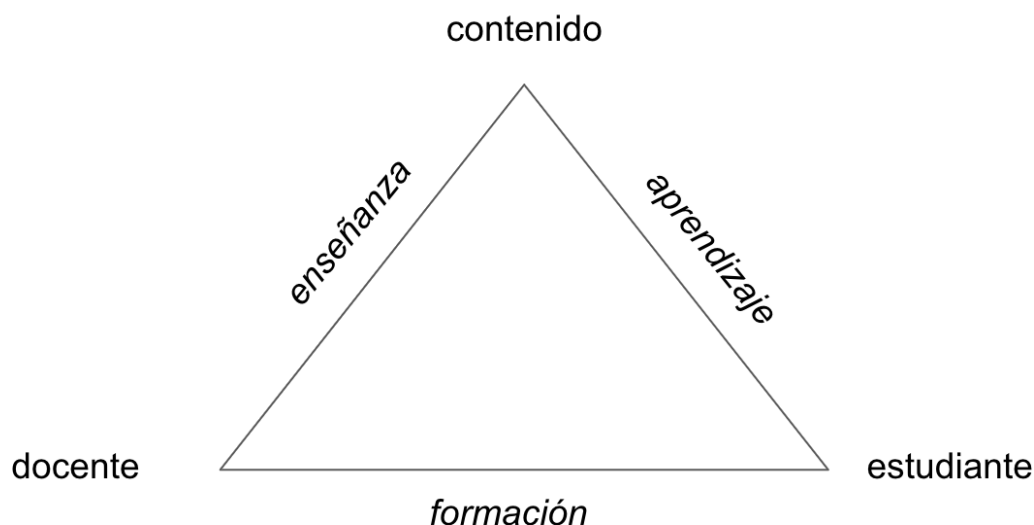


Figura 4.2: Triángulo pedagógico de Houssaye

Utilizando un modelo de enseñanza es posible definir distintos métodos de enseñanza a partir de especificar los roles de cada uno de los componentes del modelo y las relaciones entre ellos.

En términos más formales cada método de enseñanza constituye una estrategia didáctica la cual en cierto modo implica una concepción de la enseñanza y un conjunto de técnicas de enseñanza características. La didáctica tradicional considera el aprendizaje como una acción de repetir y memorizar sin mayor reflexión ni cuestionamiento lo cual hace que la capacidad de memorizar del estudiante juegue un rol decisivo en su aprendizaje. Por ello, podría decirse que está claramente centrada en el docente. La relación docente-estudiante es estrictamente jerárquica, el docente es un transmisor de información y mediador entre el estudiante y el conocimiento. El estudiante asume un rol completamente pasivo: recibe y acumula información, realizando ejercicios de manera mecánica repitiendo los procedimientos indicados por el docente. Una típica técnica de enseñanza usada en esta didáctica es la clase magistral donde el docente expone los contenidos y la comunicación es unilateral. La evaluación está típicamente basada en exámenes que representan la culminación del proceso de enseñanza y aprendizaje. Este modelo de enseñanza tiene sus raíces en el modelo pedagógico conductista del cual Burrhus Frederic Skinner (Skinner, 1988) es uno de sus máximos referentes.

Por el contrario la denominada “didáctica alternativa o integradora” está centrada en el estudiante el cual es guiado por el docente para la construcción del conocimiento. Busca que el estudiante puede llegar a la esencia del contenido y pueda establecer nexos y relaciones que le permitan solucionar situaciones no sólo en el ámbito académico sino también en la sociedad en general. Es por esto que la didáctica integradora propicia la valoración personal de lo que se estudia para que el contenido adquiriera sentido para el estudiante y este interiorice su significado. Este modelo de enseñanza está más cercano a los modelos pedagógicos cognitivista (Ausubel et al., 1989) y constructivista (Rivero, 2009; Vygotski et al., 1995)

4.2. Tecnologías de la Información y de la Comunicación

En los últimos 15 años hemos asistido a un crecimiento exponencial de la presencia de la tecnologías de la información y de la comunicación (TICs) en la vida cotidiana. El acceso masivo de la sociedad a dispositivos con conectividad a Internet ha abierto la puerta a redes sociales complementando o incluso reemplazando algunas tecnologías anteriores como el correo electrónico, el chat, y los foros de discusión. Esta situación caracterizada ya en 2005 como la Web 2.0, convirtió a la mayoría de consumidores de Internet de simple lectores/consumidores de información en potenciales productores de información. De esta forma crece no solo la cantidad de información disponible sino también la velocidad con la que esta se genera, evoluciona y envejece. Esta oferta abrumadora de información nos transforma como sociedad y genera nuevos dilemas. El ámbito educativo y los procesos de enseñanza/aprendizaje no están exentos de este impacto.

Según nos propone Coll (Coll, 2009) las claves para comprender y valorar el impacto de las TICs sobre la educación, no está en las propias TICs ni en sus características, sino en las actividades que llevan a cabo docentes y estudiantes gracias a las posibilidades de comunicación, intercambio, acceso y procesamiento de la información que les ofrecen las TICs.

Las TICs vienen a jugar un rol de mediador entre los actores del triángulo pedagógico. Ese rol puede tomar distintas formas:

- Las TICs como instrumentos mediadores de las relaciones entre los estudiantes y los contenidos.
- Las TICs como instrumentos mediadores de las relaciones entre los docentes y los contenidos.
- Las TICs como instrumentos mediadores de las relaciones entre los docentes y los estudiantes o incluso entre los estudiantes.
- Las TIC como instrumentos mediadores de la actividad conjunta desplegada por docentes y estudiantes durante la realización de las tareas o actividades de enseñanza y aprendizaje.
- Las TICs como instrumentos configuradores de entornos o espacios de trabajo y de sus aprendizaje.

De acuerdo a Zangara (Zangara, 2014) el concepto de mediación subyace a la práctica educativa y como tal, siempre ha estado presente. La particularidad es que podemos definir las prácticas educativas como prácticas mediadas por tecnología.

Resulta imposible listar todas las herramientas que las TICs pueden aportar a la educación, en parte porque son muchas y en parte porque todo el tiempo están surgiendo nuevas herramientas. Sin embargo es posible clasificar las herramientas de acuerdo a su uso. Es así como tenemos:

- Herramientas de producción de contenidos: entran en esta categoría las herramienta de edición de video, audio, texto, etc, así como también las herramientas que permiten generar paquetes de contenido educativo en formatos específicos como SCORM. Posiblemente Microsoft Power Point sea una de las herramientas más utilizadas de esta categoría en ámbitos educativos.
- Herramientas de almacenamiento: para almacenamiento de archivos genéricos como Dropbox y Google Drive o para almacenamiento de archivos de formato específico como Vimeo (video), Soundcloud (audio) y Flickr (fotos).
- Herramientas de comunicación: aquí tenemos una amplica gama que incluye desde servicios de correo electrónico, hasta servicios de chat como Slack y servicios de video conferencia como Zoom, pasando por foros de discusión y listas de correo entre otros.

- Redes sociales: algunas como Facebook, Instagram y Twitter que además sus características de conexión entre personas tiene también la posibilidad de actuar como herramientas de algunas de las otras categorías.

4.3. Entornos Virtuales

Los tipos de herramientas mencionados en la sección previa son en cierto modo de uso genérico en el sentido de que se trata de herramientas que no fueron diseñadas particularmente para la enseñanza. Complementariamente a ellas existen los denominados Entornos Virtuales de Enseñanza y Aprendizaje (EVEAs). Estos entornos, también llamados “Campus Virtuales”, son espacios virtuales que permiten implementar una propuesta de enseñanza a partir de proveer espacios de interacción, almacenamiento de contenidos y herramientas de comunicación, seguimiento y gestión académica. Estas plataformas permiten crear y gestionar espacios denominados en forma genérica como “Aulas Virtuales” las cuales pueden representar una materia, un curso o una comisión. Entre las funcionalidades que ofrecen los EVEAs suelen encontrarse:

- Almacenamiento y publicación de contenido.
- Foros de discusión.
- Chat.
- Mensajería.
- Cuestionarios para implementar actividades de evaluación.
- Soporte de “Tareas” que los estudiantes deben completar y entregar.
- Control de asistencia.
- Gestión de notas.

En la actualidad existen diversos EVEAs con distintos modelos de licenciamiento, instalación y soporte, pero hay uno que se destaca entre todos ellos: Moodle ⁵. Moodle es una plataforma gratuita y de código abierto. Es posible descargarla e instalarla en servidores propios o utilizarla como servicio en la nube. La versión 1.0 de Moodle data de

⁵<https://moodle.org/>

2001. A marzo de 2020 la página de oficial de Moodle ⁶ reporta 157.000 sitios basados en Moodle y un total de más de 195 millones de usuarios. Entre las características que han ayudado a su gran popularidad destacan: sus capacidades de personalización, su constante evolución, su diseño modular y su inmensa comunidad. Algunos otros EVEAs populares en la actualidad son: Google Classroom ⁷, Educativa ⁸, Instructure Canvas ⁹ y TalentLMS ¹⁰.

4.4. Modalidades de enseñanza

A partir de la frecuencias de encuentro presencial y temporal de docente y estudiante es posible definir distintas modalidades de enseñanza. Es así que surge la denominada Educación a Distancia la cual se posiciona como una modalidad educativa en la que los docentes y estudiantes no tienen una frecuencia de encuentro en un lugar físico determinado para llevar adelante el proceso enseñanza y aprendizaje. Esta modalidad de enseñanza surgió a mediados del siglo XX con el fin de compensar la imposibilidad que tenían las personas que vivían alejadas de los centros educativos urbanos para asistir regularmente a clase. Obviamente, desde el surgimiento de la educación a distancia hasta la actualidad la situación a cambiado mucho y en diversas dimensiones. Por un lado, los conceptos de enseñanza y aprendizaje han sido enriquecidos con nuevas miradas. Por otro lado, la masividad de las tecnologías de la información y de la comunicación han colaborado a “acortar distancias”. Es así que Michael Moore (Moore and Kearsley, 2011) nos habla de distancia transaccional en lugar de distancia geográfica. Esta distancia transaccional se caracteriza por 3 variables:

- La estructura diseñada para lo que se pretende enseñar.
- El diálogo y la interacción entre el docente y el estudiante.
- El grado de autonomía del estudiante que interactúa con el docente dentro de la estructura diseñada.

⁶<https://stats.moodle.org/>

⁷<https://classroom.google.com/>

⁸<https://www.educativa.com/>

⁹<https://www.instructure.com/>

¹⁰<https://www.talentlms.com/>

Según la teoría elaborada por Moore estas tres variables determinan una distancia transaccional que es un espacio psicológico y comunicacional de potencial desentendimiento entre el docente y el estudiante. La figura 4.3 esquematiza los conceptos de la distancia transaccional de Moore.

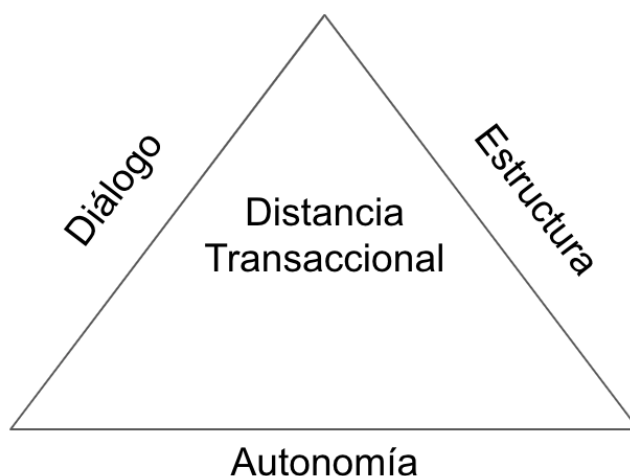


Figura 4.3: Distancia Transaccional de Moore

Tenemos entonces un continuo de modalidades educativas con la educación presencial tradicional en un extremo y la educación a distancia en el otro. Entre estos dos extremos se encuentran distintas modalidades con distinto grado de hibridación. En la actualidad es cada vez más común que independientemente del grado de hibridación de la propuesta, la implementación de la misma utilice con mayor o menor protagonismo algún Entorno Virtual de Enseñanza y Aprendizaje. A continuación se describen brevemente algunas de estas modalidades que han tomado cierta relevancia.

4.4.1. Blended Learning

El *Blended Learning* (Bonk and Graham, 2012) a veces traducido como “aprendizaje combinado” o “aprendizaje flexible”, implica una estrategia en la que se combinan las estrategias presenciales y a distancia a partir de considerar las particularidades de los estudiantes, los contenidos y demás componentes del modelo de aprendizaje. Es común que una propuesta de tipo *blended* incluya el diseño de materiales específicos para que los estudiantes consuman a distancia.

4.4.2. Bimodal

En el modelo bimodal existen dos propuestas con distintas modalidades: una basada en lo presencial y otra a distancia o en modalidad *blended* y se le permite al estudiante elegir en que modalidad cursar de acuerdo a su situación particular. El gran desafío de esta modalidad radica en cómo garantizar el mismo nivel de aprendizaje en ambas modalidades. Para esto suele utilizarse la misma bibliografía, las mismas tareas y las mismas evaluaciones.

4.4.3. Aula extendida

El aula extendida (o *extended learning* en inglés) es una variante de la educación presencial que hace uso de la tecnologías de la información para extender el aula en términos de acceso a recursos e interacción docente-estudiante entre otros. El caso más simple de aula extendida podría ser un blog o una simple página web donde el docente publica materiales de estudio de la materia. Debe tenerse presente que la propuesta es una propuesta presencial, el estudiante está obligado a asistir a la clase presencial porque es ahí donde se da (al menos administrativa y formalmente) la experiencia de enseñanza y aprendizaje.

4.4.4. Aula invertida

El aula invertida o “flipped classroom” (Milman, 2012) como se la suele denominar en inglés, es una variante de aula extendida y, al igual que esta, las actividades presenciales se complementan con actividades mediadas por la tecnología. La “inversión” propuesta por esta modalidad tiene que ver con hacer en el aula lo que tradicionalmente se hacía en casa y hacer en casa lo que tradicionalmente se hacía en el aula. Esto típicamente se traduce en que los estudiantes estudian el material teórico en sus casas, muchas veces a partir de videos o lecturas que se disponibilizan por medio de un aula virtual. Luego, en el aula los estudiantes trabajan en actividades prácticas con la guía/asistencia del docente. De esta forma el estudiante toma el centro de la escena dentro del aula en lo que suele denominarse “educación centrada en el educando” (*learner-centered education*).

5. Diseño del relevamiento

Son nuestras elecciones, Harry, las que muestran lo que somos, mucho más que nuestras habilidades.

Albus Dumbledore

Para llevar adelante el relevamiento se optó por realizar una encuesta basada en un cuestionario online. El diseño de la encuesta se hizo siguiendo los lineamientos de Linaker (Linaker et al., 2005). El cuestionario completo de la encuesta se encuentra en el apéndice 1.

La unidad de observación de la encuesta es a nivel curso/comisión/materia en la que se dictan métodos ágiles, mientras que la unidad de análisis tiene una doble interpretación dependiendo de la pregunta de investigación a la que se pretenda dar respuesta. Para las preguntas de investigación PI2 a PI5, la unidad de análisis es el curso o comisión (dependiendo de la terminología y estructura de cada institución) pero para la pregunta de investigación PI1, la unidad de análisis es la carrera. Las preguntas de investigación PI2 a PI5 tienen que ver con decisiones que puede tomar cada equipo docente. Puede que una determinada materia tenga varios cursos y/o comisiones y que cada una tenga una estrategia de enseñanza distinta, esta situación generaría distintas respuestas a cada una de las preguntas de investigación PI2 a PI5.

Se tomo especial cuidado en regular la extensión del cuestionario de forma de asegurar que el tiempo que lleve completarlo no exceda los 10 minutos para mitigar el riesgo de que los encuestados desistan de completarlo.

En el cuestionario se incluyen preguntas de tipo abierto (texto libre), cerrado (elección de opciones predefinidas) y mixto, o sea: se le presenta al encuestado preguntas con un conjunto de opciones de respuesta y dentro de esas opciones existe la posibilidad de elegir la opción otro y agregar una opción diferente a las propuestas. Asimismo, en la mayoría de las preguntas de tipo cerradas y mixtas se incluyó una opción “Desconozco”.

En las siguientes secciones se detallan diversos aspectos del diseño e implementación de la encuesta y en el apéndice 1 se presenta el cuestionario completo.

5.1. Audiencia

Como audiencia de esta encuesta se establecieron docentes de materias de Ingeniería de Software debido a que en términos curriculares los temas de métodos ágiles están dentro de la temática Ingeniería de Software. No se hizo distinción en la categoría de los docentes encuestados, o sea: no importó si el encuestado era profesor, jefe de trabajos prácticos o

ayudante. El punto relevante fue que quien contestara la encuesta fuera parte activa del equipo docente de la materia.

5.2. Implementación del cuestionario

El cuestionario de la encuesta fue implementado con la herramienta web Google Forms¹¹ la cual ofrece un conjunto de funcionalidades muy útiles para la manipulación de los datos recolectados. Entre las funcionalidades destacadas de esta plataforma está la generación automática de gráficos y la posibilidad de exportar los datos a una planilla de cálculo.

5.3. Estructura del cuestionario

El cuestionario contó con un total de 22 preguntas estructuradas en cuatro secciones:

1. **Información general:** en esta sección se pidieron datos personales de contacto para el caso de necesitar despejar alguna duda al momento de procesamiento y análisis de resultados. También, se incluyeron algunas preguntas de índole demográfica: carrera, institución y materia. Finalmente, se incluyeron instrucciones sobre cómo contestar el resto de la encuesta: las preguntas de las secciones 2 y 3 debían ser contestadas contextualizadas en una única materia.
2. **Información sobre el contenido:** esta sección indaga sobre qué métodos y prácticas ágiles son enseñadas. Concretamente, se incluyó una pregunta sobre los métodos enseñados y otra sobre las prácticas. Ambas son del tipo selección múltiple. La lista de métodos y prácticas corresponde con los ítems mencionados en las secciones 3.3 y 3.4. En ambos casos se incluyó la opción “Otros” para dar la posibilidad a los encuestados de incluir métodos y prácticas que no se hubieran tenido en cuenta en el diseño del cuestionario.
3. **Información sobre la forma de enseñanza:** esta sección contiene preguntas al respecto de estrategias didácticas, uso de tecnologías, herramientas de mediación, etc. La formulación de estas preguntas representó todo un desafío pues la formación docente no es un requisito obligatorio para ejercer la docencia en universidades

¹¹<https://www.google.com/forms/about/>

argentinas. Debido a ello, son muy pocos los docentes universitarios con formación pedagógica. Es por esto que en la redacción de las preguntas de esta sección se tomó especial cuidado de no utilizar terminología pedagógica específica que pueda resultar extraña para quienes completaran la encuesta. Adicionalmente, en algunos casos se agregaron párrafos aclaratorios de términos que se supuso podrían resultar ambiguos o no lo suficientemente claros.

4. **Cierre:** esta sección contiene dos preguntas de texto libre: una para que quien llena la encuesta pueda agregar comentarios y/o aclaraciones y otra para proveer información de contacto con otra gente que considere debería contestar la encuesta.

5.4. Evaluación preliminar del cuestionario

La evaluación del cuestionario se hizo en dos etapas. Inicialmente el cuestionario fue revisado por los directores de este trabajo de especialización en un esquema iterativo a partir del cual se realizaron mejoras incrementales. Luego en una segunda etapa se hizo una prueba piloto de la encuesta con dos colegas del autor que trabajan en distintas universidades quienes completaron el cuestionario. Esto permitió obtener feedback en dos sentidos: por un lado estos colegas dieron feedback explícito luego de completar el cuestionario y por otro lado las respuestas obtenidas se pudieron cotejar con las respuestas esperadas dado que el autor trabaja en los mismos equipos docentes que estos colegas.

5.5. Ejecución de la encuesta

Si bien la encuesta estaba publicada en internet en forma abierta, su dirección de acceso era un link autogenerado por Google forms de más de 50 caracteres de longitud y su difusión no se hizo en forma masiva. Sino que, para asegurar un alto ratio de respuesta, el link de la encuesta fue enviado individualmente por mail personalizado a docentes de Ingeniería de Software y afines de diversas universidades argentinas. Esto requirió obviamente un trabajo previo de relevamiento. El envío de mails de invitación para completar la encuesta comenzó a mediados de octubre de 2019 y se recibieron respuestas hasta fines de diciembre de 2019, momento en que se cerró el cuestionario.

6. Resultados

Todas las verdades son fáciles de entender, una vez descubiertas. El punto es descubrirlas.

Galileo Galilei

Para el relevamiento se contactaron en forma directa por correo electrónico a todas las universidades miembros de la Red UNCI. A partir de esto se consiguieron un total de 78 respuestas ya que algunos de los contactos realizados difundieron la encuesta entre colegas. Es así que incluso cuando no se consiguieron respuestas de algunas universidades de Red UNCI, se obtuvieron respuestas de universidades no pertenecientes a Red UNCI. Luego de un proceso de depuración de resultados quedaron finalmente 69 registros para el análisis. En el apéndice 2 se listan las materias/carreras/instituciones resultantes del proceso de depuración.

En el proceso de depuración se identificaron respuestas correspondientes a carreras de postgrado y pregrado que están por fuera del alcance de este trabajo y, por ello, esos registros fueron eliminados para el análisis.

Para realizar el análisis de las respuestas fue necesario trabajar tanto a nivel institución como a nivel materia/curso. Cada respuesta del cuestionario estaba enfocada en una materia lo cual habilitó que en algunos casos se tengan varias respuesta asociadas a una misma institución. Aparecen aquí algunas situaciones particulares como ser:

- Universidades que dictan más de una carrera del área de informática, como es el caso de la Universidad de Buenos Aires.
- Universidades que ofrecen varios cursos de una misma materia, ya sea porque la universidad está distribuida en varias ciudades o porque tiene varias bandas horarias. En estos casos se comparte el plan de estudios pero los docentes son potencialmente distintos. Tal es el caso de la Universidad Tecnológica Nacional.

6.1. Presencia de los métodos ágiles en las carreras

La presencia de los métodos ágiles en las carreras fue relevada ofreciendo a los encuestados un conjunto de opciones excluyentes¹². La unidad de análisis para esta pregunta fue a nivel carrera/institución. Para los casos de carreras/instituciones en los que se contó con más de una respuesta, se cotejó que las mismas fueran consistentes entre sí y en caso de no serlo se volvió a contactar a los encuestados para clarificar la situación. De esta

¹²Como en todas las preguntas de opción múltiple se permitió a los encuestados especificar una opción adicional en caso que ninguna de las provista se ajuste a sus necesidades

forma, el número total de respuestas consideradas para análisis de esta cuestión luego de la eliminación de duplicados fue de 44.

Tal como se observa en la figura 6.1 en la mayoría de los casos (54,5 %) los métodos ágiles se estudian en el contexto de varias materias que también cubren otros temas. En tan solo un caso (equivalente a 2,3 %) los métodos ágiles no son estudiados en el contexto de la carrera.

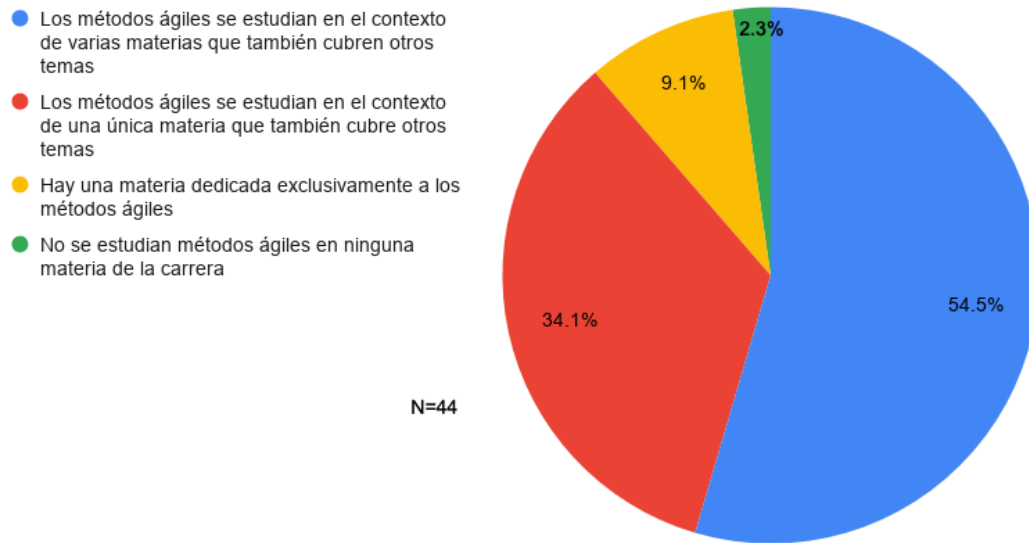


Figura 6.1: Enseñanza de métodos ágiles en las carreras

6.2. Enseñanza de métodos ágiles

En relación a los temas enseñados se relevó por separado la enseñanza de métodos ágiles y la enseñanza de prácticas ágiles. El análisis se hizo a nivel materia, ya que dentro de una misma carrera/institución distintas materias podrían cubrir distintos temas. Las figuras 6.2 y 6.3 resumen las respuestas obtenidas para la enseñanza de estas cuestiones. Fundamentos de los métodos ágiles, Desarrollo Iterativo e Incremental y *Scrum* son los métodos de desarrollo más enseñados, los tres con un porcentaje de respuestas superior al 75 %.

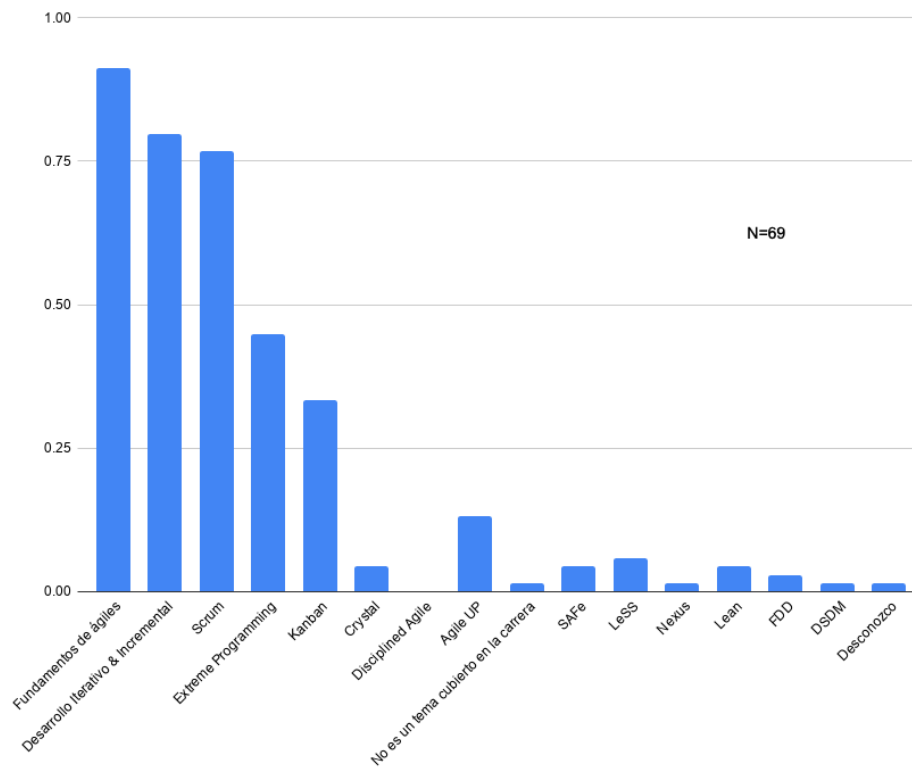


Figura 6.2: Métodos ágiles enseñados

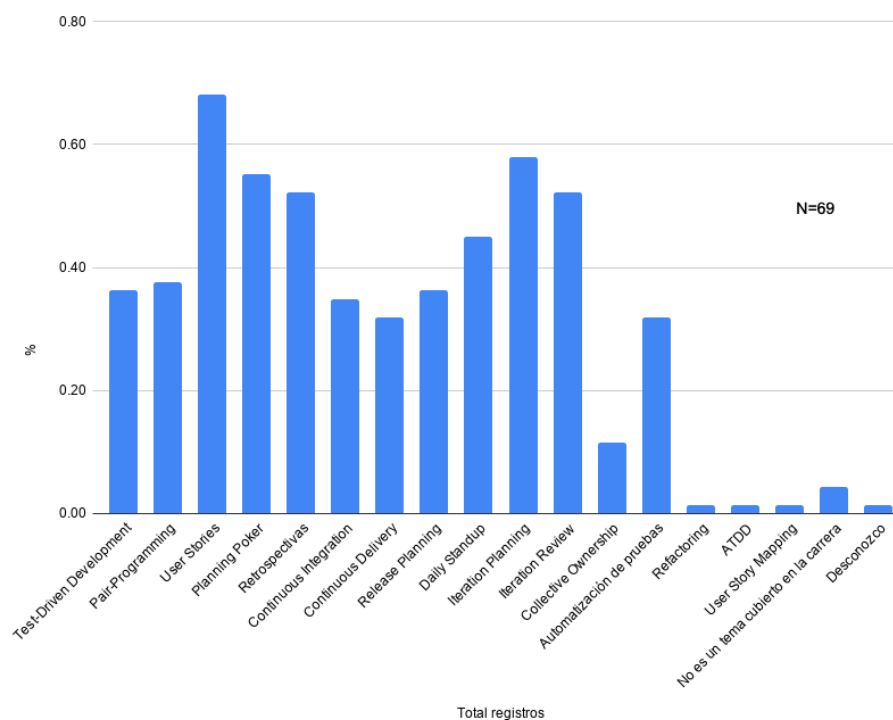


Figura 6.3: Prácticas ágiles enseñadas

Respecto de las prácticas ágiles *User Stories*, *Iteration Planning*, *Planning Pocker*, *Retrospectives*, *Iteration Review* e *Iteration Review* son las cinco prácticas con mayor porcentaje de respuestas, las cinco por encima del 50 %. Cabe destacar que estas 5 prácticas pertenecen todas a la categoría “Practicas de gestión”.

6.3. Estrategia de enseñanza

En términos de enseñanza de teoría y práctica la figura 6.4 muestra que el 31,9 % de los encuestados indicó que la enseñanza se hace en forma teórica y práctica por partes iguales, mientras que el 26,1 % indicó que predomina la enseñanza práctica sobre la teórica.

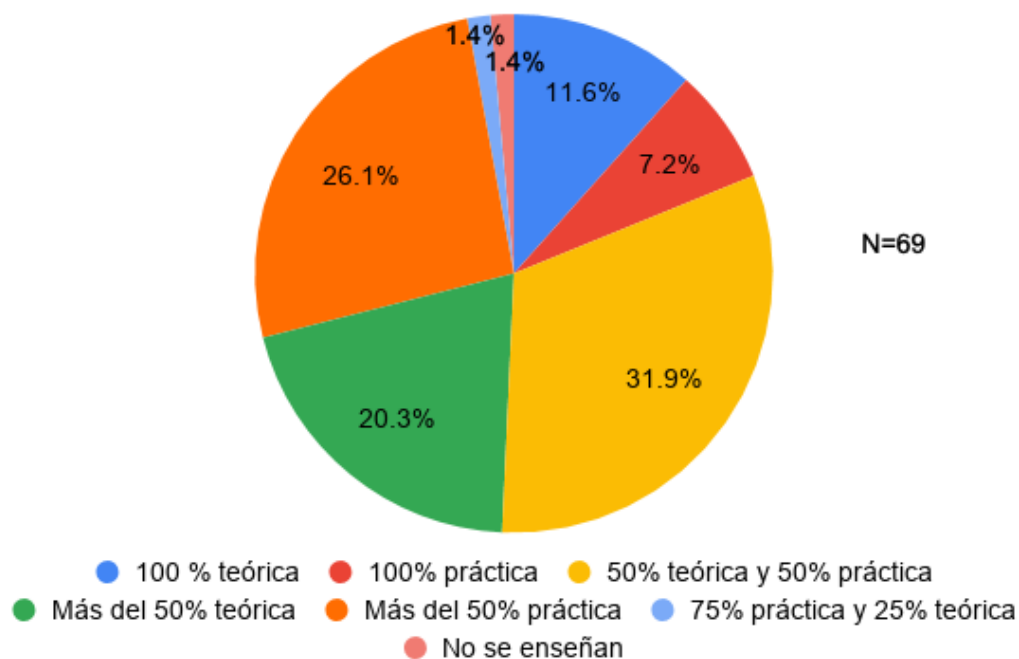


Figura 6.4: Enseñanza teórica y práctica

Respecto de la modalidad de enseñanza la gran mayoría (83,8 %) indicó que es de manera 100 % presencial mientras que el restante 16,2 % utiliza un modelo de enseñanza híbrido. Esto puede verse en la figura 6.5.

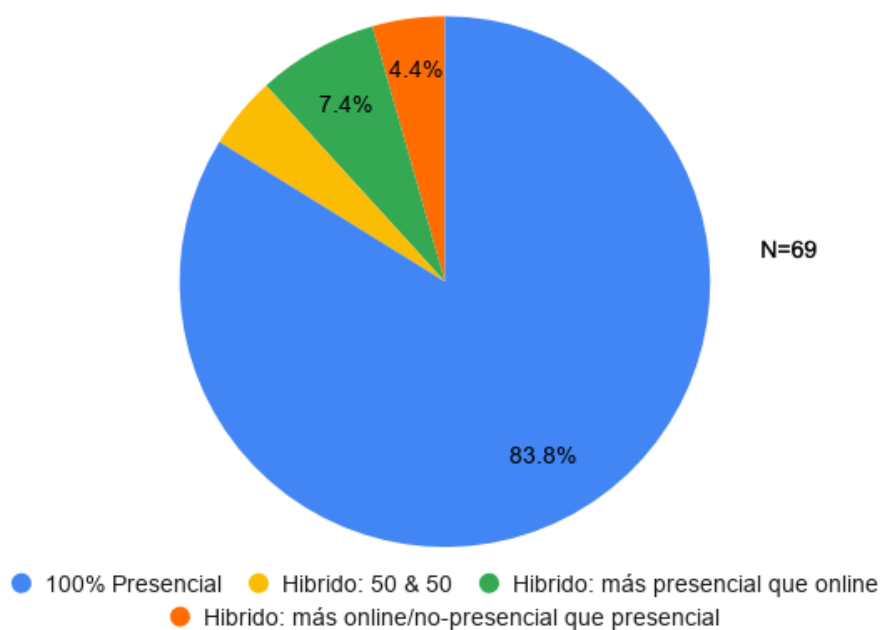


Figura 6.5: Modalidad de Enseñanza

En cuanto a la modalidad de evaluación las tareas grupales son el método más utilizado (88 %) seguido por las evaluaciones parciales (83 %). La figura 6.6 muestra el resumen de respuestas para esta pregunta.

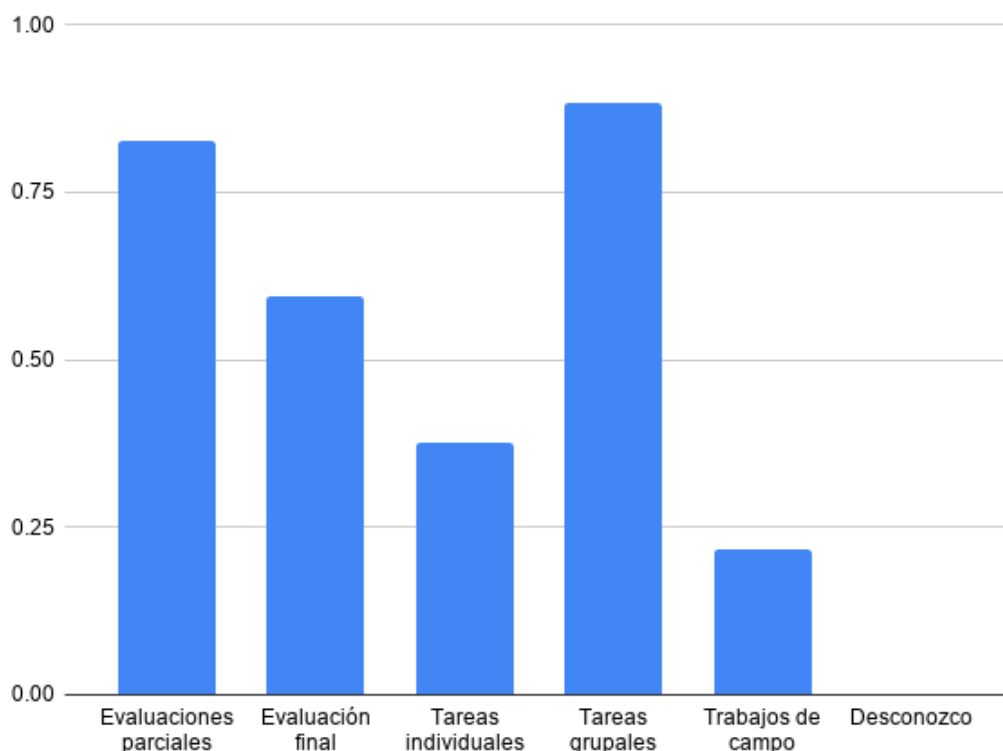


Figura 6.6: Modalidad de evaluación

6.4. Didáctica de la enseñanza

Respecto de la didáctica de la enseñanza se buscó identificar el uso de las distintas técnicas en relación al lugar de utilización:

- Dentro del aula presencial.
- Fuera del aula presencial.
- Dentro y fuera del aula presencial.
- No se utiliza la técnica en cuestión.

Los resultados indican que la exposición dialogada resultó ser la estrategia más utilizada (94 %) independientemente del lugar.

Entre las actividades realizadas exclusivamente fuera del aula se destaca el estudio de materiales (36 %).

Finalmente, al considerar las actividades que se realizan tanto fuera como dentro del aula

presencial se destacan las actividades grupales con trabajo de integración (51 %).

La tabla 6.1 muestra el detalle de las respuestas de estrategias didácticas.

Tabla 6.1: Estrategias didácticas dentro y fuera del aula (valores expresados en %)

| Estrategias Didácticas | dentro del aula | fuera del aula | dentro y fuera del aula | no se hace |
|----------------------------|-----------------|----------------|-------------------------|------------|
| Clase magistral | 32 | 0 | 4 | 64 |
| Exposición dialogada | 87 | 0 | 7 | 6 |
| Estudio de materiales | 13 | 36 | 43 | 7 |
| Act. ind. sin integración | 7 | 17 | 12 | 62 |
| Act. ind. con integración | 23 | 7 | 45 | 25 |
| Act. grup. sin integración | 12 | 6 | 20 | 62 |
| Act. grup. con integración | 35 | 0 | 51 | 14 |
| Visitas / Entrevistas | 22 | 12 | 7 | 59 |

6.5. Uso de tecnologías

Entre las herramientas tecnológicas más utilizadas se ubican en primer lugar las diapositivas (87 %) y en segundo lugar los Campus Virtuales (70 %). En tercer lugar se ubican las listas de correo (52 %). El resto de las herramientas tecnológicas tiene un uso inferior al 50 %.

La figura 6.7 resume las respuestas a esta pregunta.

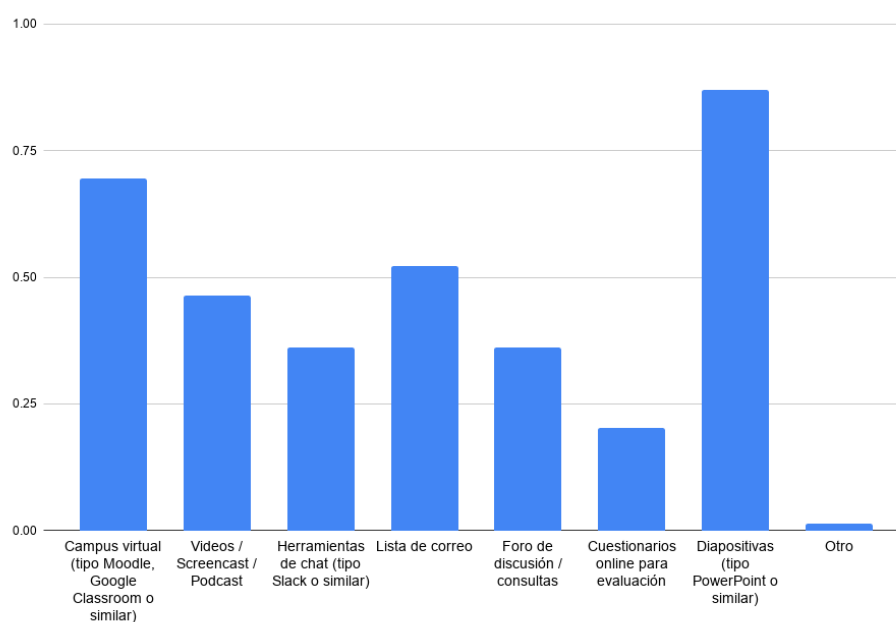


Figura 6.7: Tecnologías utilizadas en la enseñanza

6.6. Análisis cruzado de las respuestas

En las secciones previas se presentaron resultados correspondientes al análisis de las respuestas a cada una de las preguntas del cuestionario. En esta sección se analizan algunos resultados surgidos de cruzar respuestas entre preguntas.

6.6.1. Métodos versus Prácticas

Al analizar en forma conjunta las respuestas referidas a la enseñanza de métodos ágiles y la enseñanza de prácticas ágiles se evidencia una situación llamativa en los casos de *Scrum*, *Extreme Programming* y sus prácticas asociadas. Se observa que el 77 % de los encuestados indicó enseñar *Scrum*, pero tan solo el 53 % indicó enseñar Retrospectivas, que sin duda es una de las prácticas más importantes de *Scrum*. *Iteration Planning*, *Iteration Review* y *Daily Standup* son todas prácticas centrales de *Scrum* y su porcentaje de enseñanza es bastante menor al de *Scrum*.

En el caso de *Extreme Programming* se ve que es enseñando en el 44 % de los casos pero varias de sus prácticas no superan el 40 % (*Pair-Programming* 37 %, *Test-Driven Development* 36 %, *Continuous Integration* 36 % y *Collective Ownership* 11 %).

Al mismo tiempo, se observa otro fenómeno interesante: prácticas ágiles que son más enseñadas que sus respectivos métodos de origen. Este es el caso de *User Stories* que es una práctica enseñada en el 67 % de los casos, pero que su método padre, *Extreme Programming*, es solo enseñado en el 44 % de los casos.

6.6.2. Modalidad de enseñanza

El 83 % de los encuestados indicaron utilizar una modalidad completamente presencial pero al mismo tiempo el 70 % de los encuestados indicó utilizar un campus virtual. Esto sugiere que gran parte de las materias dictadas en forma presencial hacen algún tipo de extensión del aula mediante el uso de campus virtual.

7. Discusión

Es de importancia para quien desee
alcanzar una certeza en su
investigación, el saber dudar a tiempo.

Aristóteles

7.1. Sobre la representatividad de la muestra

De cara a intentar dimensionar la representatividad de la muestra con la que se trabajó se buscó información estadística de la Secretaría de Políticas Universitarias. Inicialmente se utilizó el sistema web de consulta de estadísticas de dicha secretaría¹³. Luego de realizar varias búsquedas en dicho sistema y detectar ciertas limitaciones se realizó un pedido de información en la correspondiente Página de Pedidos de Información¹⁴ del Departamento de Información Universitaria de la mencionada Secretaría. Concretamente, la información que se pidió fue la cantidad de egresados anual correspondiente a las carreras de grado del área informática desagregado por institución.

La información provista en respuesta al pedido realizado incluyó los datos correspondientes a egresados del año 2016 y 2017 para instituciones tanto de gestión pública como privada.

A partir de cruzar los datos recolectados en la encuesta es posible verificar que las instituciones que forman parte de la muestra generaron en 2017 el 77 % de los egresados de las carreras de grado de informática. El mismo porcentaje se verifica para el año 2016.

7.2. Sobre el impacto de COVID-19

El presente trabajo reporta un estado del arte a diciembre de 2019, momento en que se completó el relevamiento de datos. El trabajo de análisis y redacción se completó en abril 2020 cuando la pandemia del COVID-19 ya estaba en curso. Sin duda esta pandemia tendrá efectos en varias de las cuestiones relevadas en este trabajo, ya que varias universidades se han visto obligadas a incorporar estrategias de enseñanza virtual para el primer cuatrimestre de 2020. Habrá que analizar si a futuro estas estrategias se mantienen o no y qué impacto tienen estas nuevas estrategias sobre las metodologías de enseñanza previas, descriptas en este trabajo.

¹³<http://estadisticasuniversitarias.me.gov.ar/>

¹⁴<http://pedidosciie.siu.edu.ar/>

8. Conclusión

El viaje no termina aquí.

Gandalf el Blanco.

8.1. Sobre las preguntas de investigación

PI1: ¿Se enseñan métodos ágiles de desarrollo de software en las carreras universitarias de grado de informática en Argentina?

Esta pregunta tiene una respuesta contundente: si, los métodos ágiles son enseñados en el 97.7 % de las carreras de grado relevadas. Más aún, en el 54,5 % de los casos, los métodos ágiles se estudian en el contexto de varias materias que también abordan otros temas. Respecto de la enseñanza de prácticas ágiles cabe destacar que las 5 prácticas más enseñadas son prácticas ágiles de gestión. Esto podría, en cierto modo, justificar el hecho de que en la industria las prácticas ágiles de gestión tienen más uso que las prácticas ágiles técnicas (Paez et al., 2018; Version One, 2019).

PI2: ¿Cuál es el foco de estudio de los métodos ágiles (teórico, teórico/práctico, otros)?

En este caso no hay una estrategia mayoritaria, la estrategia más utilizada con el 31,9 % de las respuesta es enseñanza en forma teórica y práctica por partes iguales. En segundo lugar, el 26,1 % de los encuestados indicó que la enseñanza es de forma más práctica que teórica.

PI3: ¿En qué medida se utilizan tecnologías de la información en la enseñanza de métodos ágiles?

En términos del uso de tecnologías solo tres opciones superan el 50 %: las diapositivas, el campus virtual y la lista de correo. Esto delata una gran oportunidad de potencial crecimiento para la incorporación de tecnologías en el dictado de las materias.

PI4: ¿Se utiliza alguna modalidad híbrida de enseñanza como aula invertida y/o extendida?

La gran mayoría de los encuestados (83.8 %) dicta sus materias de forma completamente presencial. Este hecho puede estar influido por cuestiones de regulaciones ya que muchas instituciones no admiten alternativas de educación no presencial y exigen asistencia física a las clases. Sin embargo, el 70 % de los encuestados utiliza en sus materias un campus virtual, lo cual representa una extensión del aula y por ende cierto grado de hibridación en la modalidad de enseñanza.

PI5: ¿Qué metodologías didácticas se utilizan en la enseñanza de métodos ágiles?

La exposición dialogada es la estrategia más utilizada (94 %), lo cual resulta muy razonable considerando que la gran mayoría dicta clases completamente presenciales. El uso de actividades grupales alcanza prácticamente el 86 %, lo cual para cuestiones de métodos ágiles parece un número significativo. Uno de los puntos centrales de los métodos ágiles tiene que ver con el factor humano y la colaboración, y por ello se podría esperar un importante uso de actividades grupales en la enseñanza de estos contenidos.

8.2. Reflexiones generales

Analizando en forma conjunta las respuestas a las cinco preguntas de investigación planteadas y viendo algunos de los resultados de la encuesta que exceden las preguntas, surgen algunas cuestiones que merecen reflexión.

En términos generales, la forma de enseñanza parece ser bastante tradicional. El uso de tecnologías de la información no resulta sorprendente considerando que estamos hablando de carreras de tecnología y que los alumnos actuales de estas carreras son nativos digitales. Más aún, dadas estas particularidades se podría haber esperado a priori una mayor penetración de las tecnologías de la información en el dictado de las materias. En un punto podría considerarse paradójico que en la enseñanza de tecnologías de la información se haga un uso bastante acotado de las mismas en la forma de dictado de las materias.

Podría pensarse que, tal vez, parte de esta situación se deba al hecho que si bien los alumnos actuales son nativos digitales, posiblemente no ocurra lo mismo con la mayoría de los docentes.

8.3. Trabajos futuros

Este trabajo reporta de forma exhaustiva el estado del arte de la enseñanza de métodos ágiles en las carreras de grado del área de informática en Argentina a diciembre de 2019.

Un camino de investigación que explícitamente decidió no abordarse en este trabajo fue el

análisis de los planes de estudio de las distintas instituciones. La decisión estuvo sustentada en dos cuestiones. En primer lugar, este estudio tenía como objetivo obtener información que no suele figurar en los planes de estudio. En segundo lugar, muchas veces la duración del trámite formal de la actualización de los planes de estudio termina haciendo que los docentes hagan actualizaciones informales en los contenidos de sus materias, lo cual provoca que los planes de estudio terminen no reflejando lo que realmente se enseña. De todas formas, luego de haber completado este estudio, podría ser interesante estudiar los planes de estudio y cotejarlos con lo aquí encontrado.

Una posible línea de continuación de este trabajo podría ser la realización de casos de estudio para tener un entendimiento más profundo de algunos de los casos relevados en el presente estudio.

En otra línea de trabajo podría ser interesante desarrollar una propuesta de enseñanza enfocada particularmente en la enseñanza de métodos ágiles o al menos en algunas prácticas centrales del movimiento ágil.

Finalmente, como se mencionó en el capítulo anterior, los eventos desencadenados por el COVID-19 tendrán algún impacto en las estrategias de enseñanza. De hecho ya lo están teniendo, la cuestión será analizar si los cambios producidos por la situación de pandemia persisten en el tiempo o no. Al margen de esto la metodología de investigación utilizada en este trabajo ha probado su utilidad y puede ser replicada con ajustes para estudiar el nuevo contexto que se abrirá a partir de la pandemia. Aquí también podría resultar interesante la realización de casos de estudio para entender el proceso de transición de propuestas de enseñanza en modalidad predominantemente presenciales a modalidades completamente virtuales.

Referencias

- ACM. Acm curricula recommendations. <https://www.acm.org/education/curricula-recommendations>. [Online; accedido 2020-03-30].
- Ambler, S. and Lines, M. (2012). *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise*. IBM Press.
- Anderson, D. (2013). *Kanban*. Blue Hole Press Inc.
- Aubin, V., Blautzik, L., and Dejean, G. (2013). Mejoras en el proceso de enseñanza-aprendizaje de programación utilizando metodologías de la industria del software. In *Congreso Nacional de Ingeniería Informática / Sistemas de Información (CoNaIISI 2013)*.
- Aubin, V., Blautzik, L., Guatelli, L., Pafundi, R., Salica, F., M, N., and Zanga, Z. (2014). Metodologías activas y corrección por rúbricas en el proceso de enseñanza-aprendizaje de programación. In *Congreso Nacional de Ingeniería Informática / Sistemas de Información (CoNaIISI 2014)*.
- Ausubel, D., Novak, J. D., and Hanesian, H. (1989). *Psicología cognitiva*. México: Trilhas.
- Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. (2001). Manifesto for agile software development.
- Bonk, C. J. and Graham, C. R. (2012). *The handbook of blended learning: Global perspectives, local designs*. John Wiley & Sons.
- Burghardt, M., Lapertosa, S., Burgos, J., Vallejos, O., and G., R. (2017). La evolución de las cátedras para influir en los knowmads y formar al trabajador de la industria 4.0. In *Congreso Nacional de Ingeniería Informática / Sistemas de Información*.
- Chookittikul, W., Maher, P., and Kourik, J. (2011). Agile methods in thai higher education and beyond. page 557.
- Cockburn, A. (2004). *Crystal Clear: A Human-Powered Methodology for Small Teams*. Addison-Wesley Professional.
- Coll, C. (2009). *Aprender y enseñar en las TIC: expectativas, realidad y potencialidades en el libro Los desafíos de las TIC para el cambio educativo*. Organización de Estados Iberoamericanos para la Educación, la Ciencia y la Cultura (OEI), Servicio de Publicaciones.
- Ferraro, M. D. L. A., Gómez Solis, L., and Matoso, A. (2015). Mejora continua aplicada en la enseñanza de la ingeniería del software. In *X Congreso de Tecnología en Educación y Educación en Tecnología, TEYET 2015*.
- Gimson Saravia, L. E. and Gil, G. D. (2016). Experiencia sobre desarrollo con genexus

- siguiendo prácticas ágile. In *XI Congreso de Tecnología en Educación y Educación en Tecnología, TEYET 2016*.
- Gresse von Wangenheim, C., Savi, R., and Borgatto, A. (2013). Scrumia—an educational game for teaching scrum in computing courses. *Journal of Systems and Software*, 86:2675–2687.
- Hernández, P. (1989). *Diseñar y Enseñar, Teorías y Técnicas de la Programación del Proyecto Docente*. Narcea Ediciones.
- Houssaye, J. (1984). *Le triangle pédagogique*. PhD thesis, ANRT.
- Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. 2.
- Kropp, M. and Meier, A. (2013). Teaching agile software development at university level: Values, management, and craftsmanship. pages 179–188.
- Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Garousi, V., Felderer, M., Trektore, K., Mccaffery, F., Linssen, O., Hanser, E., and Prause, C. (2017a). Software and system development in practice: Waterfall, scrum, and beyond.
- Kuhrmann, M., Fernández, D. M., and Daneva, M. (2017b). On the pragmatic design of literature studies in software engineering: an experience-based guideline. *Empirical Software Engineer*, 22.
- Lanza Casteli, S. (2017). ¿cómo pueden scrum y las tics mejorar el aprendizaje colaborativo en el ambito universitario? In *Congreso Nacional de Ingeniería Informática / Sistemas de Información(CoNaIISI 2017)*.
- Larman, C. and Vodde, B. (2016). *Large-Scale Scrum: More with LeSS*. Addison-Wesley Professional.
- Leffingwell, D. (2018). *SAFe 4.5 Reference Guide: Scaled Agile Framework for Lean Enterprises*. Addison-Wesley Professional.
- Levy, S. J., Romero Dapozo, J. R., and Pasini, A. C. (2016). Mejora continua aplicada en la enseñanza de la ingeniería del software. In *XIX Concurso de Trabajos Estudiantiles (EST 2016)*.
- Linaker, J., Sulaman, S., Host, M., and de Mello, R. (2005). Guidelines for conducting surveys in software engineering.
- Lu, B. and Declue, T. (2011). Teaching agile methodology in a software engineering capstone course. *Journal of Computing Sciences in Colleges*, 26:293–299.
- Mahnic, V. (2012). A capstone course on agile software development using scrum. *Education, IEEE Transactions on*, 55:99 – 106.
- Melo, C., Santos, V., Katayama, E., Corbucci, H., Prikladnicki, R., Goldman, A., and Kon, F. (2013). The evolution of agile software development in brazil. *Journal of the Brazilian Computer Society*, 19.
- Milman, N. B. (2012). The flipped classroom strategy: What is it and how can it best be used? *Distance learning*, 9(3):85.

- Moore, M. and Kearsley, G. (2011). *Distance Education: A Systems View of Online Learning*. Cengage Learning, 3 edition.
- Paasivaara, M., Heikkilä, V., Lassenius, C., and Toivola, T. (2014). Teaching students scrum using lego blocks. *36th International Conference on Software Engineering, ICSE Companion 2014 - Proceedings*.
- Paez, N. (2017). A flipped classroom experience teaching software engineering.
- Paez, N., Fontdevila, D., Gainey, F., and Oliveros, A. (2018). Technical and organizational agile practices: A latin-american survey. In Garbajosa, J., Wang, X., and Aguiar, A., editors, *Agile Processes in Software Engineering and Extreme Programming*, pages 146–159, Cham. Springer International Publishing.
- Paez, N., Oliveros, A., Fontdevila, D., and Zangara, M. A. (2019). Introducing agile methods in undergraduate curricula, a systematic mapping study. In *XXV Congreso Argentino de Ciencias de la Computación (CACIC)(Universidad Nacional de Río Cuarto, Córdoba, 14 al 18 de octubre de 2019)*.
- Persson, M., Kruzela, I., Alder, K., Johansson, O., and Johansson, P. (2012). On the use of scrum in project driven higher education.
- Pressman, R. (2014). *Software Engineering: A Practitioner’s Approach*. McGraw-Hill Education.
- Resolución 786/2009 (2009). Ministerio de Educación de la República Argentina. <http://servicios.infoleg.gob.ar/infolegInternet/verNorma.do?id=154121>. [Online; accedido 2020-03-30].
- Rico, D. and Sayani, H. (2009). Use of agile methods in software engineering education. pages 174–179.
- Rivero, M. (2009). Teoría genética de piaget: Constructivismo cognitivo. *Universitat de Barcelona*. Obtenido de [http://diposit.ub.edu/dspace/bitstream/2445/32321/6/Teoria % 20de % 20Jean % 20Piaget. pdf](http://diposit.ub.edu/dspace/bitstream/2445/32321/6/Teoria%20de%20Jean%20Piaget.pdf).
- Rodríguez, G., Soria, A., and Campo, M. (2012). Teaching scrum to software engineering students with virtual reality support.
- Rodríguez, G., Soria, A., and Campo, M. (2015a). Supporting assessment of practices in software engineering courses. *IEEE Latin America Transactions*, 13:3142–3148.
- Rodríguez, G., Soria, A., and Campo, M. (2015b). Virtual scrum: A teaching aid to introduce undergraduate software engineering students to scrum. *Computer Applications in Engineering Education*, 23.
- Rodríguez, G., Soria, A., and Campo, M. (2016). Measuring the impact of agile coaching on students’ performance. *IEEE Transactions on Education*, 59:1–1.
- Sacristán, G. (1983). *Teoría de la enseñanza y desarrollo del currículo*. Anaya.
- Schwaber, K. (1997). Scrum development process. In Sutherland, J., Casanave, C., Miller, J., Patel, P., and Hollowell, G., editors, *Business Object Design and Implementation*, pages 117–134, London. Springer London.

- Scott, E., Rodríguez, G. H., Soria, A., and Campo, M. (2013). El rol del estilo de aprendizaje en la enseñanza de prácticas de scrum: Un enfoque estadístico. In *XIV Simposio Argentino de Ingeniería de Software (ASSE)-JAIIO 42*.
- Scott, E., Rodríguez, G., Soria, A., and Campo, M. (2014). Are learning styles useful indicators to discover how students use scrum for the first time? *Computers in Human Behavior*, 36:56–64.
- Scott, E., Rodríguez, G., Soria, A., and Campo, M. (2016). Towards better scrum learning using learning styles. *Journal of Systems and Software*, 111:242–253.
- Scott, E., Rodríguez, G., Soria, A., and Campo, M. (2017). *Experiences in software engineering education: Using scrum, agile coaching, and virtual reality*, pages 1257–1283.
- Skinner, B. F. (1988). *The selection of behavior: The operant behaviorism of BF Skinner: Comments and consequences*. CUP Archive.
- Sommerville, I. (2015). *Software Engineering*. Pearson.
- Tranter, L. (2018). Are agile and scrum the same? <https://www.extremeuncertainty.com/are-agile-and-scrum-the-same/>. [Online; accedido 2020-03-30].
- Uva, M., Daniele, M., Zorzán, F. A., Frutos, M., and Arsaute, A. (2014). Propuesta para documentar trabajos finales utilizando metodologías ágiles. In *IX Congreso de Tecnología en Educación y Educación en Tecnología (TEYET 2014)*.
- Version One (2019). 13th Annual State of Agile Report. <https://www.stateofagile.com>. [Online; accedido 2020-03-30].
- Vygotski, L. S., Kozulin, A., and Abadía, P. T. (1995). *Pensamiento y lenguaje*. Paidós Barcelona.
- West, D. and Grant, T. (2010). Agile development: Mainstream adoption has changed agility.
- Wijnands, W. and Stolze, A. (2019). *Transforming Education with eduScrum: Bringing Methodologies from Industry to the Classroom*, pages 95–114.
- Zangara, M. A. (2014). Apostillas sobre los conceptos básicos de educación a distancia o... una brújula en el mundo de la virtualidad. *Material didáctico del Seminario de Educación a Distancia, Maestría en “Tecnología Informática aplicada en Educación Facultad de Informática, UNLP*.

Apéndice

A1. Apéndice 1 - Cuestionario

Encuesta sobre la enseñanza de métodos ágiles en universidades de Argentina (v2)

Esta encuesta es parte de un estudio realizado en el contexto de la Maestría en Tecnología Informática Aplicada en Educación de la Universidad Nacional de La Plata.

El objeto del estudio es establecer los contenidos y métodos de enseñanza de los enfoques ágiles en las carreras de informática de las universidades argentinas.

Dicho estudio es llevado a cabo por el Ing. Nicolás Paez bajo la dirección la Dra. Alejandra Zangara y del Lic. Alejandro Oliveros y el asesoramiento técnico de Mg. Diego Fontdevila.

La información provista en esta encuesta será tratada de forma confidencial y los resultados del trabajo serán compartidos una vez finalizado el mismo.

Este cuestionario tiene 3 secciones principales:

1. Información general
2. Información sobre contenidos enseñados
3. Información sobre la forma de enseñanza

Para contestar las secciones 2 y 3 le pedimos que tome como referencia 1 materia en particular

* Required

Información general

1. d1. Nombre y Apellido *

2. d2. Mail *

3. d3. Institución en la que da clases *

Si dicta clases en más de una institución, por favor complete un formulario por cada institución.

4. d4. Carrera en la que da clases *

Si dicta clases en más de una carrera en la misma Institución, por favor complete un formulario por Carrera. Si no hay diferencias relevantes entre las carreras entonces basta con que lo mencione en la pregunta X1 al final del cuestionario.

Mark only one oval.

- ☐ Licenciatura en Ciencias de la Computación
- ☐ Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas
- ☐ Licenciatura en Informática
- ☐ Ingeniería en Computación
- ☐ Ingeniería en Sistemas de Información/Informática
- ☐ Other: _____

5. d5. Materias en las que se enseñan métodos ágiles en el contexto de la carrera *

Mark only one oval.

- ☐ Hay una materia dedicada exclusivamente a los métodos ágiles
- ☐ Los métodos ágiles se estudian en el contexto de una única materia que también cubre otros temas
- ☐ Los métodos ágiles se estudian en el contexto de varias materias que también cubren otros temas
- ☐ No se estudian métodos ágiles en ninguna materia de la carrera
- ☐ Desconozco
- ☐ Other: _____

6. d6. Materia en la que da clases y en base a la cual contestará el resto de este cuestionario *

Información sobre Contenidos

Las preguntas en esta sección son referentes a la materia identificada en la pregunta d6

7. a1. Enfoques ágiles que se enseñan en la materia en cuestión (marque todos los que aplican) *

Check all that apply.

- ☐ Fundamentos de los métodos ágiles en general
- ☐ Desarrollo Iterativo & Incremental
- ☐ Scrum
- ☐ Extreme Programming
- ☐ Kanban
- ☐ Crystal
- ☐ Disciplined Agile
- ☐ Agile UP
- ☐ No es un tema cubierto en la carrera
- ☐ SAFe
- ☐ LeSS
- ☐ Desconozco

Other: ☐ _____

8. a2. Prácticas ágiles que se enseñan en la materia en cuestión (marque todos los que aplican) *

Check all that apply.

- ☐ Test-Driven Development
- ☐ Pair-Programming
- ☐ User Stories
- ☐ Planning Poker
- ☐ Retrospectivas
- ☐ Continuous Integration
- ☐ Continuous Delivery
- ☐ Release Planning
- ☐ Daily Standup
- ☐ Iteration Planning
- ☐ Iteration Review
- ☐ Collective Ownership
- ☐ Automatización de pruebas
- ☐ No es un tema cubierto en la carrera
- ☐ Desconozco

Other: ☐ _____

Información sobre la forma de enseñanza

Las preguntas en esta sección son referentes a la materia identificada en la pregunta d6

9. e1. La enseñanza de métodos ágiles se realiza de forma teórico/práctica en la siguiente proporción *

Mark only one oval.

- ☐ 100 % teórica
- ☐ 100% práctica
- ☐ 50% teórica y 50 % práctica
- ☐ Más del 50% teórica
- ☐ Más del 50% práctica
- ☐ Desconozco
- ☐ Other: _____

10. e2. Modalidades de enseñanza y aprendizaje *

Mark only one oval.

- ☐ 100% Presencial
- ☐ 100% No-presencial / Online (videos, clases remotas por internet con herramientas como skype, slack, etc))
- ☐ Híbrida: 50% no-presencial/online + 50% presencial
- ☐ Híbrida: más del 50% presencial + no-presencial/online
- ☐ Híbrida: más del 50% no-presencial/online + presencial
- ☐ Desconozco
- ☐ Other: _____

11. e3. Para cada una de las siguientes opciones indicar: a) si ocurre dentro del aula, b) ocurre fuera del aula presencial o c) no se hace. Entendiendo aula como aula presencial donde alumnos y docentes están juntos físicamente *

Mark only one oval per row.

| | dentro del aula | fuera del aula | dentro y fuera del aula | no se hace |
|---|-----------------------|-----------------------|----------------------------|-----------------------|
| Clase magistral: el docente expone y la participación de los alumnos es mínima y de carácter pasivo | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Exposición dialogada: el docente expone pero estimula mediante preguntas la participación activa de los alumnos | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Estudio de materiales (lecturas, video, etc) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Actividades individuales SIN actividad de integración/puesta en común posterior | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Actividades individuales CON actividad de integración/puesta en común posterior | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Actividades en pequeños grupos SIN actividad de integración/puesta en común | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Actividades en pequeños grupos CON actividad de integración/puesta en común posterior | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Visitas / Entrevistas a profesionales | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

12. e4. Dinámica de evaluación (marque todas las que aplican) *

Check all that apply.

- ☐ Evaluaciones parciales
- ☐ Evaluación final
- ☐ Tareas individuales
- ☐ Tareas grupales
- ☐ Trabajos de campo
- ☐ Desconozco

Other: ☐ _____

13. e5. Tecnologías de la información utilizadas en el dictado de la materia (marque todas las que aplican) *

Check all that apply.

- ☐ Campus virtual (tipo Moodle, Google Classroom o similar)
- ☐ Videos / Screencast / Podcast
- ☐ Herramientas de chat (tipo Slack o similar)
- ☐ Lista de correo
- ☐ Foro de discusión / consultas
- ☐ Cuestionarios online para evaluación
- ☐ Diapositivas (tipo PowerPoint o similar)
- ☐ Desconozco

Other: ☐ _____

Cierre

14. x1. Por favor escriba cualquier comentario que considere necesario mencionar respecto de la enseñanza de métodos ágiles

15. x2. ¿Hay algún otro docente que considere debería contestar esta encuesta? En caso positivo por favor proveanos su información de contacto

A2. Apéndice 2 - Participantes del relevamiento

A continuación se listan en orden alfabético las asignaturas que participaron del relevamiento junto con la carrera e institución perteneciente.

- Administración de Proyectos. Ingeniería en Sistemas de Información/Informática. Universidad del CEMA
- Administración de Proyectos. Ingeniería en Sistemas de Información/Informática. Universidad Nacional del Sur
- Administración y Control de Proyectos Informáticos. Ingeniería en Sistemas de Información/Informática. Universidad de Buenos Aires
- Análisis de sistemas de información. Ingeniería en Sistemas de Información/Informática. Universidad de Morón
- Diseño de Aplicaciones. Ingeniería en Sistemas de Información/Informática. Universidad de Morón
- Elementos de Ingeniería de Software. Licenciatura en Informática. Universidad Nacional de Hurlingham
- Gestión de Proyectos de Software. Ingeniería en Sistemas de Información/Informática. Universidad CAECE
- Gestión de Proyectos de Software. Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas. Universidad Nacional de la Patagonia Austral (UA Río Gallegos)
- Ingeniería de Software. Ingeniería en Sistemas de Información/Informática. Universidad Nacional de La Matanza
- Ingeniería de Software. Ingeniería en Computación. Universidad Nacional de Tres de Febrero
- Ingeniería de Software. Ingeniería en Sistemas de Información/Informática. Universidad Austral
- Ingeniería de Software. Licenciatura en Informática. Universidad Atlántida Argentina

- Ingeniería de Software. Ingeniería en Sistemas de Información/Informática. Universidad Nacional del Chaco Austral
- Ingeniería de Software. Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas. Universidad Nacional de Salta
- Ingeniería de Software. Ingeniería en Sistemas de Información/Informática. Universidad Nacional de Avellaneda
- Ingeniería de Software. Ingeniería en Sistemas de Información/Informática. Universidad Nacional de Avellaneda
- Ingeniería de Software. Ingeniería en Computación. Universidad Nacional de Mar Del Plata
- Ingeniería de Software. Licenciatura en Informática. Universidad Nacional de la Patagonia San Juan Bosco (Sede Madryn)
- Ingeniería de Software. Licenciatura en Ciencias de la Computación. Universidad Nacional de Río Cuarto
- Ingeniería de Software. Ingeniería en Sistemas de Información/Informática. Universidad Tecnológica Nacional (FR Buenos Aires)
- Ingeniería de Software. Ingeniería en Sistemas de Información/Informática. Universidad Tecnológica Nacional (FR Tucumán)
- Ingeniería de Software. Ingeniería en Sistemas de Información/Informática. Universidad Tecnológica Nacional (FR Córdoba)
- Ingeniería de Software 1. Ingeniería en Sistemas de Información/Informática. Instituto Tecnológico de Buenos Aires
- Ingeniería de Software 1. Licenciatura en Ciencias de la Computación. Universidad Nacional de San Luis
- Ingeniería de Software 1. Ingeniería en Computación. Universidad Nacional de Tucumán
- Ingeniería de Software 1. Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas. Universidad Nacional de Tierra del Fuego

- Ingeniería de Software 1. Ingeniería en Sistemas de Información/Informática. Universidad de la Cuenca del Plata
- Ingeniería de Software 1. Licenciatura en Ciencias de la Computación. Universidad de Buenos Aires
- Ingeniería de software 1. Ingeniería en Sistemas de Información/Informática. Universidad Nacional de Catamarca
- Ingeniería de Software 1. Licenciatura en Ciencias de la Computación. Universidad Nacional de Córdoba
- Ingeniería de Software 1. Licenciatura en Informática. Universidad Nacional del Oeste
- Ingeniería de Software 1. Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas. Universidad Nacional de Río Negro
- Ingeniería de Software 1 y 2. Licenciatura en Ciencias de la Computación. Universidad Nacional de Rosario
- Ingeniería de Software 2. Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas. Universidad Nacional de Río Negro
- Ingeniería de Software 2. Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas Universidad Nacional de La Plata
- Ingeniería de Software 1. Ingeniería en Sistemas de Información/Informática. Universidad Atlántida Argentina
- Ingeniería de Software 1. Ingeniería en Sistemas de Información/Informática. Universidad de la Marina Mercante
- Ingeniería de Software 1. Ingeniería en Sistemas de Información/Informática. Universidad de Belgrano
- Ingeniería de Software I-T. Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas. Universidad Nacional de la Patagonia San Juan Bosco (Sede Trelew)

- Ingeniería de software 2. Ingeniería en Sistemas de Información/Informática. Pontificia Universidad Católica Argentina
- Ingeniería de Software 3. Ingeniería en Sistemas de Información/Informática. Universidad Nacional de Catamarca
- Ingeniería del Software. Ingeniería en Sistemas de Información/Informática. Universidad Tecnológica Nacional (FR Tucumán)
- Ingeniería del Software 3. Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas. Universidad Nacional de Misiones
- Ingeniería del Software y Técnicas de Desarrollo de Software ágiles. Ingeniería en Sistemas de Información/Informática Universidad Tecnológica Nacional (FR Resistencia)
- Ingeniería en Software. Ingeniería en Sistemas de Información/Informática. Universidad Tecnológica Nacional (FR Buenos Aires)
- Metodología de Sistemas 2. Licenciatura en Informática. Universidad de Entre Ríos
- Metodologías Ágiles. Ingeniería en Sistemas de Información/Informática. Universidad Tecnológica Nacional (FR La Plata)
- Metodologías Ágiles en Desarrollo de Software. Ingeniería en Sistemas de Información/Informática. Universidad Tecnológica Nacional (FR Rosario)
- Metodologías Avanzadas. Ingeniería en Sistemas de Información/Informática. Universidad Gaston Dachary
- Metodologías de Análisis de Sistemas. Ingeniería en Sistemas de Información/Informática. Universidad Abierta Interamericana
- Métodos Ágiles para Aplicaciones Web. Licenciatura en Informática. Universidad Nacional de La Plata
- Métodos Ágiles para el Desarrollo de Software. Ingeniería en Sistemas de Información/Informática. Universidad Nacional del Centro de la Provincia de Buenos Aires

- Métodos Ágiles para el diseño de Software. Ingeniería en Sistemas de Información/Informática. Universidad Tecnológica Nacional (FR Santa Fe)
- Métodos y Modelos en Ingeniería de Software 1. Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas. Universidad de Buenos Aires
- Métodos y Modelos en la Ingeniería de Software 2. Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas. Universidad de Buenos Aires
- Planificación y gestión de sistemas de información. Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas. Universidad Nacional de la Patagonia San Juan Bosco (Sede Madryn)
- Práctica Profesional. Licenciatura en Informática. Universidad Nacional del oeste
- Proyecto Final. Ingeniería en Sistemas de Información/Informática. Universidad Tecnológica Nacional (FR Córdoba)
- Proyectos de Software. Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas. Universidad Nacional de Lanús
- Reingeniería de Procesos y de Sistemas de Información. Ingeniería en Sistemas de Información/Informática. Universidad Nacional de Catamarca
- Requerimientos de Software. Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas. Universidad Nacional de la Patagonia Austral (UA Río Gallegos)
- Sistemas de Información. Ingeniería en Sistemas de Información/Informática. Universidad Adventista del Plata
- Sistemas de Información 2. Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas Universidad Nacional de Jujuy
- Sistemas de Información 3. Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas. Universidad Nacional de Luján
- Taller de Desarrollo de Proyectos 2 Ingeniería en Sistemas de Información/Informática Universidad de Buenos Aires

- Taller de Métodos Ágiles. Ingeniería en Sistemas de Información/Informática. Universidad Nacional del Centro de la Provincia de Buenos Aires
- Elementos de Ingeniería de Software. Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas. Universidad Nacional de Quilmes
- Tópicos Avanzados de Ingeniería del Software. Licenciatura en Sistemas/Sistemas de Información/Análisis de Sistemas. Universidad Nacional del Nordeste
- Trabajo final (título intermedio). Licenciatura en Ciencias de la Computación. Universidad Nacional de Río Cuarto